Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Genome Analysis

chainCleaner improves genome alignment specificity and sensitivity

Hernando G. Suarez[1,2], Bjoern E. Langer[1,2], Pradnya Ladde[1,2] and Michael Hiller[1,2]*

[1]Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany; [2]Max Planck Institute for the Physics of Complex Systems, Dresden, Germany

*To whom correspondence should be addressed: hiller@mpi-cbg.de

Associate Editor: Dr. John Hancock

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Accurate alignments between entire genomes are crucial for comparative genomics. However, computing sensitive and accurate genome alignments is a challenging problem, complicated by genomic rearrangements.

Results: Here we present a fast approach, called chainCleaner, that improves the specificity in genome alignments by accurately detecting and removing local alignments that obscure the evolutionary history of genomic rearrangements. Systematic tests on alignments between the human and other vertebrate genomes show that chainCleaner (i) improves the alignment of numerous orthologous genes, (ii) exposes alignments between exons of orthologous genes that were masked before by alignments to pseudogenes, and (iii) recovers hundreds of kilobases in local alignments that otherwise would fall below a minimum score threshold. Our approach has broad applicability to improve the sensitivity and specificity of genome alignments.

Availability: http://bds.mpi-cbg.de/hillerlab/chainCleaner/ or https://github.com/ucscGenomeBrowser/kent

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Whole genome alignments are of fundamental importance for comparative genomic studies. Alignments between genomes of different species have been used to detect sequence conservation, which is a hallmark of purifying selection and identifies genomic regions with a biological function (Cooper, et al., 2005; Siepel, et al., 2005). Comparative genomics discovered that the majority of the evolutionarily conserved DNA sequences does not code for proteins (Birney, et al., 2007; Hillier, et al., 2004; Siepel, et al., 2005; Waterston, et al., 2002) and that a sizeable portion of these conserved regions originated from exapted transposon insertions (Lindblad-Toh, et al., 2011; Lowe, et al., 2007). Specific sequence conservation patterns were further used to predict the type of function that is encoded in this genomic region, which contributed to completing catalogs of coding and non-coding genes, uncovered conserved transcription factor binding sites and regulatory elements, and detected regions where different molecular functions overlap (Lin, et al., 2011; Lindblad-Toh, et al., 2011; Stark, et al., 2007). Furthermore, genome alignments are not only key to detect sequence similarity between species, but also to detect genomic differences that underlie phenotypic changes (Hiller, et al., 2012; McLean, et al., 2011; Pollard, et al., 2006; Prabhakar, et al., 2006; Prudent, et al., 2016). Finally, alignments are the basis to infer the evolutionary history of genomes by reconstructing ancestral genomes (Blanchette, et al., 2004a; Ma, et al., 2006).

The accuracy of genome alignments critically affects the results of the comparative analysis. However, computing genome alignments is a complex and challenging computational task (Earl, et al., 2014). While standard sequence alignment only considers substitutions, insertions and deletions, genome alignment must additionally deal with genomic rearrangements such as duplications, translocations and inversions (Dewey, 2012). A number of different genome alignment approaches have been developed (Angiuoli and Salzberg, 2011; Blanchette, et al., 2004b; Bray and Pachter, 2004; Brudno, et al., 2003a; Brudno, et al., 2003b; Darling, et al., 2010; Dubchak, et al., 2009; Frith and Kawaguchi, 2015; Grabherr, et al., 2010; Kent, et al., 2003; Paten, et al., 2011a; Paten, et al., 2011b; Paten, et al., 2008) that can be grouped into two main classes, hierarchical and local approaches (Dewey, 2012). Hierarchical alignment approaches first determine orthologous segments between the genomes that lack rearrangements and then align the bases in these segments. Local approaches first determine all local alignments and then filter them to find orthologous alignments.

© The Author (2017). Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

The chain and net framework belongs to the class of local approaches (Kent, et al., 2003). The UCSC genome browser group routinely uses chains and nets to compute pairwise alignments between a reference and a query genome (Rosenbloom, et al., 2015). Several pairwise alignments between a reference and different query species can be combined into a multiple alignment with MULTIZ (Blanchette, et al., 2004b). This chain/net/MULTIZ pipeline has been used to create many multiple genome alignments, including an alignment between human and 99 other vertebrates (Rosenbloom, et al., 2015).

To build alignment chains, co-linear local alignments (aligning blocks) that occur in the same order on a reference and a query chromosome are "chained" together (Kent, et al., 2003). Aligning blocks in such a chain can be separated by big insertions or deletions, such as transposon insertions that happened in either the reference or query. Aligning blocks can also be separated by genomic regions where the reference and query sequence does not align to each other. For example, such an un-aligning region can be a diverged intron that is flanked by conserved exons, which overlap aligning blocks. Finally, aligning blocks can be separated by regions where the reference or the query genome underwent a rearrangement, such as an inversion or translocation. Alignments of the rearranged region are not co-linear with up- and downstream aligning blocks of this chain, however they can form a separate chain.

While chains represent alignments to orthologous and paralogous regions in general, alignment nets attempt to capture only orthologous alignments. Nets are a hierarchical collection of chains or parts of chains that are organized in different levels. To build this hierarchical structure, one starts by taking all aligning blocks of the top-scoring chain as the level 1 net. Then, entire or parts of lower-scoring chains can fill the non-aligning regions (gaps) in the top-level net, becoming a level 2 net. This procedure iterates until all chains are processed (Kent, et al., 2003). Thus, while a single locus in the reference genome can overlap aligning blocks of several different chains, a locus can overlap at most one aligning block of a single net. The hierarchy of nets should ideally represent the order and number of the genomic rearrangements, which allows reconstructing the evolutionary rearrangement history (Ma, et al., 2006). For example, an inversion breaks co-linearity and results in two nested nets. The first (level 1) net aligns the locus upstream and downstream of the inversion and has a gap overlapping the inverted region. The inverted region would align in the reverse orientation in the second (level 2) net that fills this gap in the level 1 net (Figure 1A). Similarly, a translocation would align as a second level net. However, the requirement that a locus in the reference can overlap at most one net implies that nets cannot represent duplications in the query genome (Kent, et al., 2003).

A key feature of the net-building algorithm is that it takes all aligning blocks of the top-scoring chain as the top-level net. This implies that if the top-level chain contains, for example, non-orthologous alignments between the reference and the query genome, these alignments will become aligning blocks in the top-level net. Since nested lower-scoring chains can only fill gaps in a higher-scoring net, the nested chain could be broken into a number of smaller nets (Figure 1). This can lead to situations where nets do not represent the correct rearrangement history, for example by inflating the number of rearrangements that occurred (Figure 1).

In the following, we refer to aligning blocks in a top-level "breaking chain" that break a nested lower-scoring chain ("broken chain") as chain-breaking alignments (CBAs, Figure 2A). We define two types of CBAs. *True CBAs* need to be removed from the breaking chain in order to result in nets representing the correct rearrangement history (with the limitation that nets cannot represent duplications in the query genome). These removed alignments then form a new chain and can become a new nested net. In contrast, *false CBAs* should not be removed from the breaking chain, because this chain results in a net that already represents the correct rearrangement history.

Apart from obscuring the rearrangement history, true CBAs can have other undesirable consequences. First, true CBAs can mask alignments between exons of orthologous genes, for example, if the breaking chain contains alignments to a processed pseudogene. In the case shown in Figure 1C, the pseudogene alignments reveal numerous gene-inactivating mutations, from which one would incorrectly infer gene loss (Supplementary Figure 2). Second, since low scoring nets are less likely to represent an orthologous alignment, one often filters out nets with a score below a minimum threshold (Kent, et al., 2003). Consequently, if the broken chain is broken into a number of smaller nets, some of these individual nets can fall below the score threshold and would be incorrectly filtered out. This is shown in Figure 1C, where several orthologous aligning blocks are missed in the final genome alignment. Together, true CBAs impair both the specificity and sensitivity of genome alignments.

Given that the accuracy of genome alignments is crucial for comparative genomics, we developed a fast method, called chainCleaner, to detect and remove true CBAs from the breaking chains. chainCleaner relies on a score ratio that accurately distinguishes true from false CBAs. We systematically tested this method on vertebrate genome alignments at a variety of evolutionary distances and show that chainCleaner improves the alignment of many orthologous genes and rescues nets that would otherwise be incorrectly filtered out.

2 Methods

2.1 chainCleaner detects and removes true CBAs

Our method chainCleaner takes a set of alignment chains as input and removes CBAs from these chains. The rationale of chainCleaner is the following: If the part of the broken chain that surrounds the CBA represents an orthologous alignment and a single rearrangement, then the local score of the broken chain should be higher than the score of the CBA (Figure 2B). This is in contrast to the scores of the entire chains, where, by definition, the breaking chain scores higher than the broken chain. Using the chain-scoring scheme developed in (Kent, et al., 2003), we compute the score of the CBA and the scores of the broken chain parts upstream and downstream of the CBA (Figure 2B). Then, we obtain the ratio between the minimum score of the upstream/downstream broken chain parts and the score of the CBA. For true CBAs, this ratio should be > 1, while it should be < 1 for false CBAs. Given the set of chains, chainCleaner computes the score ratio for every observed CBA and removes those CBAs where the score ratio is above a certain threshold.

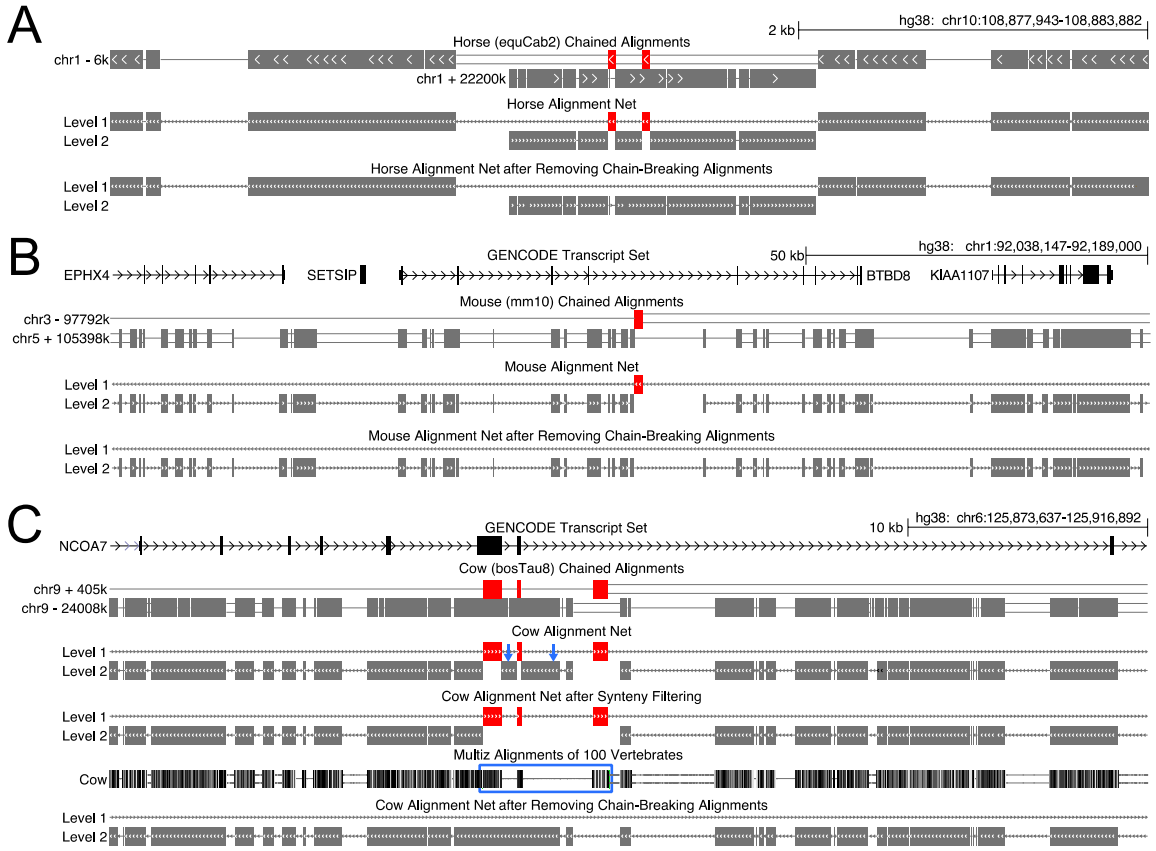

Fig. 1 Chain-breaking alignments (CBAs) in mammalian genome alignments.
UCSC genome browser illustrations show GENCODE genes and alignment chains/nets between human and horse, mouse or cow. Boxes in the chains/nets represent local aligning blocks.
(A) A genomic inversion results in two overlapping chains. Two CBAs (highlighted in red) break the lower level chain, representing a single inversion event, into three separate nets, which would imply that three inversion events happened. Both CBAs are very low scoring (lastz score <2400) and thus are likely random alignments that just arise by chance (Supplementary Figure 1). Removing both CBAs results in a single net that correctly indicates a single inversion.
(B) A chain-breaking alignment in the top-level chr3 chain breaks the lower level chr5 chain, representing a 4.1 Mb translocation, into two separate nets. Removing this chain-breaking alignment results in a single net, which spans the full *BTBD8* and its neighboring genes. The red block aligns a retroposed *GAPDH* pseudogene that likely was inserted independently into this locus in both human and mouse.
(C) Several CBAs break the chr9 minus strand chain that aligns *NCOA7* to its ortholog in cow. These CBAs align parts of *NCOA7* to a putative nuclear receptor coactivator pseudogene in cow. The orthologous alignments of two *NCOA7* exons are masked by these pseudogene alignments, which harbor numerous gene-inactivating mutations (Supplementary Figure 2). For the MULTIZ genome alignment, alignment nets were filtered for strong "syntenic alignments" (netFilter –syn from the UCSC source code (Kent, et al., 2003)), which removes two of the incorrectly broken nets (blue arrows). As a result, the MULTIZ alignment contains gene-to-pseudogene alignments and misses orthologous alignments (blue box). Removing the CBAs would keep the entire lower level chain as one syntenic net.

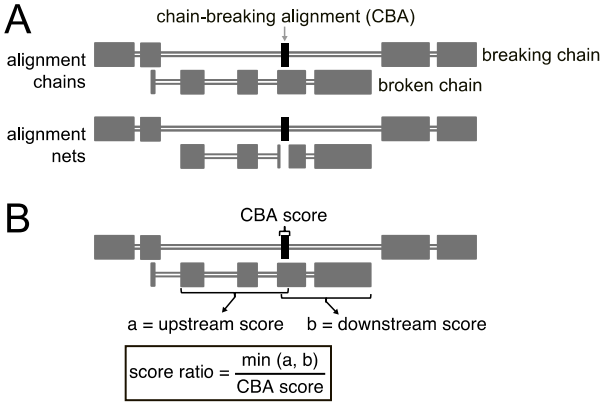

Fig. 2 Terminology and illustration of the score ratio. (A) Scheme: The "breaking chain" contains an isolated local alignment that breaks the "broken chain" into several nets. (B) Illustration of the ratio between the minimum score of the upstream and downstream broken chain parts and the score of the CBA. We use a modified scoring scheme analogous to a local alignment score (always ≥0) to score the CBA.

2.2 chainCleaner implementation

chainCleaner was implemented in C using data structures and functions from the UCSC Kent source code that efficiently handle chains and nets (Kent, et al., 2003). As a result, chainCleaner runs typically less than 10 minutes for mammalian-sized genomes, and less than 3 minutes for smaller genomes. chainCleaner requires as input a chain file and the genome sequences (2bit format).

First, chainCleaner nets the chains using chainNet and removes all individual nets with a score lower than 3000. Then it parses the fill and gap lines of the nets and uses the chain identifier (id field) to obtain a linked list of "breakInfo" objects. These objects store the identifiers, pointers to the breaking and broken chain, the coordinates of the CBA and the coordinates of non-aligning regions (gaps) upstream and downstream of the CBA. The latter coordinates correspond to the two regions where parts of the broken chain fill the gaps in the net that corresponds to the breaking chain. chainCleaner does not consider cases where the individual nets resulting from a single broken chain are at different levels since these

cases typically involve more than one breaking chain. It also does not consider cases where the entire breaking chain is nested inside the broken chain, as the resulting score ratio would be < 1 by definition (the CBA score then equals the score of the entire breaking chain). To assure that the broken chain likely represents an orthologous alignment, chainCleaner only considers broken chains with a score higher than 50000. For mammalian alignments, where chain scores are generally higher, we used 75000 as a threshold (parameter -minBrokenChainScore 75000).

Then chainCleaner loops over all CBAs, and uses the chain-scoring scheme developed in (Kent, et al., 2003) to compute the score of the CBA and the score of the part of the broken chain in the gap upstream and downstream of the CBA (see Figure 2B). This scoring scheme iterates over all aligning blocks and adds the scores of all local ungapped alignments and a cost that penalizes the gap between two adjacent blocks depending on the gap size in the reference and query assembly (Kent, et al., 2003). We noticed that a CBA can comprise several aligning blocks spread over a larger region. The score of the CBA can then be negative, for example if the CBA comprises one solid and one weak aligning block that are separated by a large distance. To avoid underestimating the score of the CBA, we scored CBAs with a modified scoring scheme that is analogous to a local alignment score. This modified scheme also iterates over all aligning blocks and but records the maximum and sets the score to 0 if it falls below 0. Then, we compute the ratio between the minimum score of the upstream and downstream broken chain parts and the score of the CBA. If this score ratio is above a user-given threshold (2.5 by default, which gives consistently high precision and sensitivity; see Results), chainCleaner removes the CBA from the breaking chain. By default, chainCleaner does not consider CBAs that score higher than 100000.

For each removed CBA, a new chain is created that gets a new chain ID. This new chain can become a new net if it fills a gap and is above a minimum score threshold. Since a breaking chain can have more than one CBA in close proximity, chainCleaner updates the size of the upstream and downstream gap in the breakInfo structures and iteratively tests if further CBAs should be removed. In addition, chainCleaner also tests if a pair of CBAs should be removed together (parameter -doPairs). Considering pairs allows removing CBAs that are very close to each other, in which case the score of the upstream or downstream part of the broken chain would not be very high (Supplementary Figure 5). We recompute the chain score for all breaking chains where CBAs have been removed. The output of chainCleaner is a cleaned and score-sorted chain file, and a file in bed format that lists the coordinates and information of each removed CBA.

2.3 Alignments between exons of orthologous genes

To determine the score ratio threshold, we downloaded the coordinates of Ensembl coding genes from the UCSC genome browser "ensGene" table for human (hg38 assembly), horse (equCab2), cow (bosTau4), mouse (mm10), opossum (monDom5), platypus (ornAna1), chicken (galGal4), lizard (anoCar2), frog (xenTro3) and zebrafish (danRer10). We used liftOver to map these genes from bosTau4 to bosTau8, ornAna1 to ornAna2 and xenTro3 to xenTro7. For testing chainCleaner on independent species, we used rat (rn6), guinea pig (cavPor3), rabbit (oryCun2), dog (canFam3), Tasmanian devil (sarHar1), zebra finch (taeGut2), duck (anaPla1), Chinese softshell turtle (pelSin1), fugu (fr3) and medaka (oryLat2). One-to-one orthologs were downloaded from Ensembl Biomart (Kinsella, et al., 2011). Then, we tested for all aligning blocks in all chains if a block aligns an exon of a human gene to its ortholog in the query species. For each human exon for which this was the case, we obtained the coordinates and the chain identifier.

2.4 Training set of true and false CBAs

We used chainCleaner with parameter –suspectDataFile to obtain the coordinates and score ratios of all chain-breaking alignments, without removing any of them and without considering pairs of CBAs. Then we overlapped all CBAs with coordinates of the genic regions (defined as the region between the first and last coding exon with an orthologous alignment for this gene) and the coordinates of the exons that align to the ortholog. As illustrated in Figure 3B, a true CBA overlaps the genic region and breaks the orthologous chain. A false CBA overlaps an alignment between exons of orthologous genes and breaks a lower-level chain that is unlikely to represent an orthologous alignment (Figure 3C).

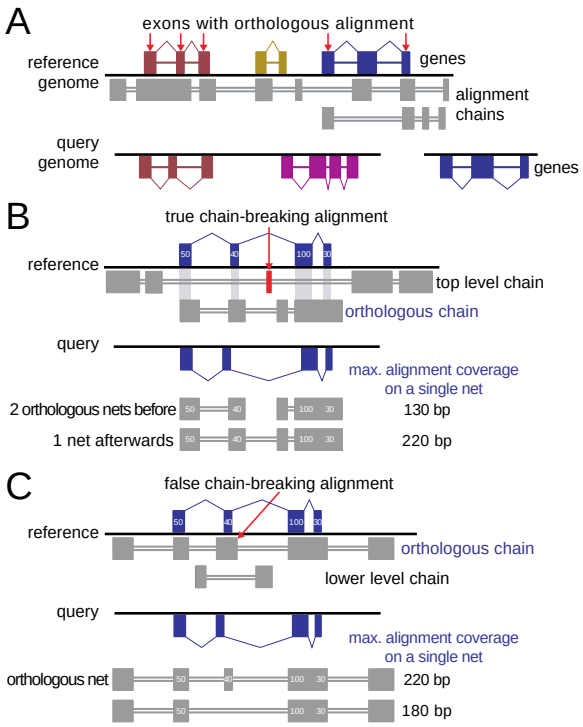

Fig 3: Exons that align between orthologous genes are used to obtain a training set of true and false chain-breaking alignments.
(A) Illustration of exons that align between orthologous genes: Genes with the same color are 1:1 orthologs. Only coding exons are considered. The top-level chain aligns the three exons of the red gene to its ortholog, however this chain also aligns exon 2 and 3 of the blue gene to a potential paralog (purple). The lower level chain aligns exon 1 and 3 of the blue gene to its ortholog. Note that exon 2 of the blue gene and exon 1 of the yellow gene align, but neither of them align to the ortholog.
(B) A CBA that is located between the first and last exon of a gene and breaks the chain that represents the orthologous alignment (lower level chain here) is considered to be a true CBA. Before removing this CBA, the orthologous exon alignments of the blue gene are located on two separate nets. After removing this CBA, all orthologous exon alignments are located on a single net, which increases the maximum number of aligning exonic bases (alignment coverage) observed for a single net.
(C) An alignment between an exon of an orthologous gene that breaks a lower level chain is considered to be a false CBA. In this case, the top-level chain represents the ortholog alignment. Removing this CBA decreases the alignment coverage.

2.5 Pairwise genome alignments

The human hg38 genome assembly was used as the reference genome. The examples in Figure 1 show chains and nets as computed by the UCSC genome browser group. For all other tests, we computed pairwise alignment chains/nets using lastz (Harris, 2007) version 1.03.54 and

doBlastzChainNet.pl (part of UCSC Kent source code) with default parameters (chainMinScore 1000, chainLinearGap loose). We integrated running chainCleaner and highly-sensitive local alignments (Hiller, et al., 2013) into the doBlastzChainNet.pl pipeline script. To align placental mammals, we used the alignment parameters K=2400 L=3000 Y=9400 H=2000 and the lastz default scoring matrix (HoxD70). To align non-placental mammals, we used K=2400 L=3000 Y=3400 H=2000 and the HoxD55 scoring matrix. For non-placental mammals, we also used highly-sensitive local alignments (Hiller, et al., 2013) with lastz parameters K=1500 L=2500 and W=5 to find co-linear alignments in the un-aligning regions that are spanned by local alignments (gaps in the chains). We filtered all local alignments for a minimum alignment quality by keeping only those alignments where at least one ≥30 bp region has ≥60% sequence identity and ≥1.8 bits of entropy as described in (Hiller, et al., 2013).

2.6 Filtering nets

We used chainNet (Kent, et al., 2003) to obtain nets from a set of chains. However, chainNet approximates the score of "sub-nets" (nets that come from a part of a chain and fill a gap in a higher-level net) by the fraction of aligning bases. While this approximation is overall quite accurate, it can lead to a bias in case the aligning blocks of a chain are not equally distributed. Therefore, we implemented a new parameter –rescore in chainNet that computes the real score of each subnet. The netFilter program (Kent, et al., 2003) filters nets according to specified criteria and applies a recursive filtering, which removes all nested nets if their parent is removed. We found that in certain cases recursive filtering removes high-scoring nested nets (Supplementary Figures 19, 20). Therefore, we implemented a non-nested filtering procedure that considers and filters each net individually, adjusting the net level in case a parent net is removed but not a net nested within.

3 Results

3.1 The score ratio distinguishes true from false CBAs

In order to distinguish true from false CBAs based on a score ratio threshold, we need a labeled training set of CBAs. Given that the true evolutionary history is generally unknown for real genomes, we included only alignment chains that span coding genes in our training set, because coding genes have three desirable properties. First, gene orthology can be determined independent of genome alignments. Here, we only used coding genes that are annotated as 1:1 orthologs in Ensembl (Herrero, et al., 2016). While the gene tree based approaches used to determine 1:1 orthology relationship are accurate, these annotations are inferred and thus are not the ground truth (see Supplementary Figure 9). Second, many coding exons align even over large evolutionary distances (Clarke, et al., 2012), which implies that truly orthologous alignments can be obtained also for distant species. Third, conserved genes maintain a co-linear exon order, which results in chains where aligning blocks represent conserved exons.

For each coding exon in the reference species, we determined whether it overlaps an aligning block of a chain that aligns this exon to an exon in the annotated 1:1 ortholog (Figure 3A). This chain is called "orthologous chain" with respect to this gene. A CBA that breaks an orthologous chain between the first and last aligning exon of the orthologous gene is considered to be a true CBA. As illustrated in Figure 3B, such a CBA results in two separate nets that align exons of the 1:1 ortholog. Removing this CBA would lead to a single net that spans the entire gene. A CBA that aligns an exon to an exon of the 1:1 ortholog and breaks a lower level chain is considered to be a false CBA. As illustrated in Figure 3C, such a CBA is an orthologous alignment and should not be removed. In order to determine score ratio thresholds that can be applied to alignments between species of various evolutionary distances, we computed alignment chains between human (reference) and the following nine query genomes: horse, cow, mouse, opossum, platypus, chicken, lizard, frog and zebrafish. These query species cover different clades within the vertebrates and their evolutionary distance to human ranges from 0.32 (horse) to 2.2 (zebrafish) substitutions per neutral site.

We found that the score ratio distribution is significantly different between true and false CBAs (Wilcoxon test: $P < 9e\text{-}15$ for all species pairs; Figure 4). Using the ratio to classify CBAs, we obtained an area under the Receiver Operating Characteristics curve (AUC) of ≥0.89 for all species, except the human-zebrafish pair (AUC 0.76, Figure 4). The lower performance for human-zebrafish is a consequence of the additional whole genome duplication that happened in teleosts (Amores, et al., 1998), which produces many paralogous chains that differentially lost genes and complicates orthology assignment (see Supplementary Figure 9). Nevertheless, even for human-zebrafish, a sensitivity of 55% at a high specificity of 98% can be achieved. We conclude that the score ratio clearly distinguishes true and false CBAs.

Given that there are many more false than true CBAs, we searched for a score ratio threshold that achieves a high precision. Precision is defined as the proportion of true CBAs of all CBAs that exceed the defined threshold and would be removed by chainCleaner. We found that a score ratio threshold of 2.5 consistently achieves a high precision for all nine species pairs (Figure 4). Overall, using this threshold, 97% of the CBAs that chainCleaner removes are true CBAs (97% precision) and 76% of all true CBAs in our training set are removed (76% sensitivity). Since false CBAs overlap exons and true CBAs are mostly intronic, we further simulated the evolution of a 3 Mb genomic segment without any genes to exclude the possibility that the score ratio mainly distinguishes coding from non-coding alignments. Using a threshold of 2.5, we obtained a sensitivity of 65% and a high precision of 92% (Supplementary Figure 14). Therefore, we used a threshold value of 2.5 for all subsequent analyses.

3.2 chainCleaner improves nets representing orthologous gene alignments

After applying chainCleaner to the chains of the nine species pairs, we determined the effect of removing CBAs on the alignments of 1:1 orthologs. We defined alignment coverage of a given gene as the sum of all exonic bases that align to the ortholog in a single net. Then, we determined the maximum alignment coverage by considering all individual nets that align exons of this gene to its ortholog. Figure 3B illustrates the case where a true CBA breaks the orthologous chain into two nets, each aligning ~50% of the total exonic bases. Removing this CBA will lead to a single net that aligns all exons, which increases the maximum alignment coverage. In contrast, removing a false CBA reduces the maximum alignment coverage (Figure 3C). Therefore, if chainCleaner correctly removes true CBAs, the maximum alignment coverage should increase for many genes.

As shown in Table 1A, chainCleaner improved the maximum alignment coverage of a total of 447 genes (examples are shown in Supplementary Figures 3-9), while decreasing the coverage of only nine genes (Supplementary Figures 9-13). To test chainCleaner on species that have not been used above for determining the score ratio threshold, we aligned 10 additional vertebrates and found that the maximum alignment coverage increased for 326 and decreased for only three genes (Table 1B). This validates that chainCleaner achieves a high precision in removing true CBAs, which improves nets that represent orthologous gene alignments.

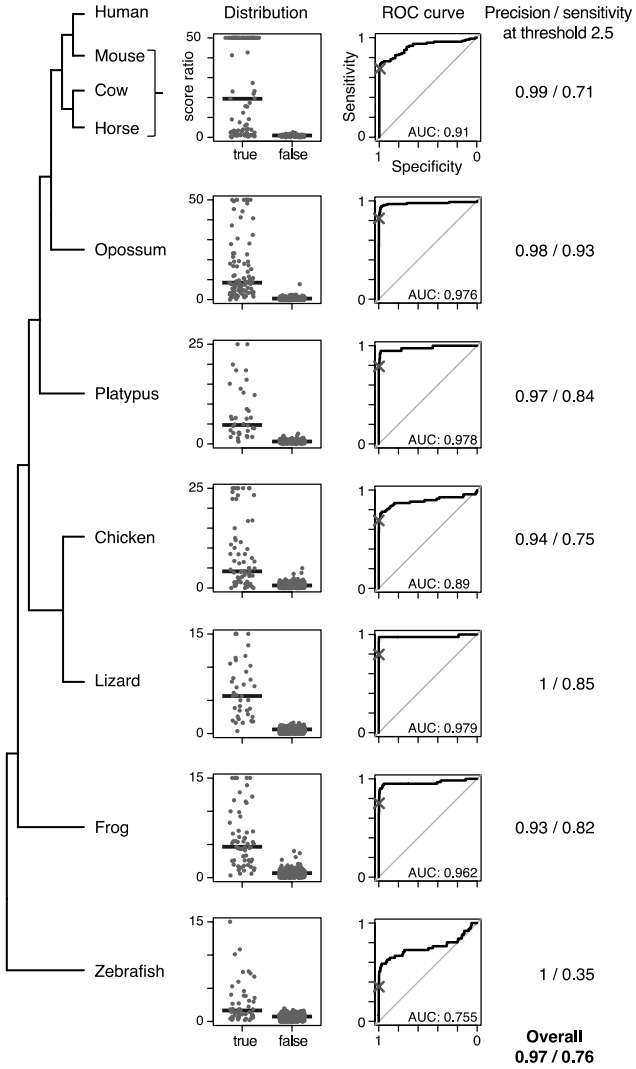

Fig. 4: The score ratio distinguishes true from false chain-breaking alignments. The score ratio differs significantly between true and false CBAs. One-dimensional scatterplots show the score ratio of the CBAs in our training set, the horizontal line is the median. For clarity of visualization, we capped very high score ratios at a maximum value to avoid plotting ratios up to 15,639 for cow. Receiver operating characteristics (ROC) curves compare the sensitivity achieved at a certain specificity. The area under the ROC curves show that this score ratio distinguishes true from false CBAs. ROC curves were plotted with pROC (Robin, et al., 2011). The cross indicates the performance at a score ratio threshold of 2.5.

3.3 chainCleaner results in individual nets with more aligning orthologous genes

Until here we have only considered CBAs that are located in genic regions between the first and last coding exon of a gene. However, true CBAs also occur in intergenic regions, where they can break chains, which align orthologous genes that occur in a conserved order in both species. Removing these CBAs is expected to lead to longer nets with a higher number of aligning orthologs (illustrated in Supplementary Figure 15). To confirm this, we compared nets before and after applying chainCleaner and determined the maximum number of orthologous genes that align on individual nets. We used alignments between exons of 1:1 orthologs to assure that only orthologous gene alignments were considered. Applying chainCleaner to the alignments of the human genome to nine other vertebrates, we found that the maximum number of aligning orthologs increased for a total of 203 nets (Supplementary Figure 16) and decreased for only three nets (Table 2, Supplementary Figures 17). This shows that chainCleaner results in nets with a higher number of aligning orthologous genes.

A

Query species	Number of genes where max. alignment coverage increases	decreases
Horse	25	0
Cow	44	1
Mouse	42	0
Opossum	82	2
Platypus	62	1
Chicken	67	2
Lizard	40	0
Frog	57	2
Zebrafish	28	1
Sum	**447**	**9**

B

Rat	51	1
Guinea pig	32	1
Rabbit	33	0
Dog	31	0
Tasmanian devil	18	0
Zebra finch	71	1
Duck	23	0
Chinese softshell turtle	29	0
Fugu	17	0
Medaka	21	0
Sum	**326**	**3**

Table 1: Number of genes where the maximum alignment coverage increases or decreases after applying chainCleaner for species used for training (A) and testing (B). Alignment coverage is the sum of all exonic bases that align to the ortholog in a single net (illustrated in Figure 3B and C).

Query species	Number of nets where number of aligning orthologs increases	decreases
Horse	22	0
Cow	37	0
Mouse	27	0
Opossum	53	0
Platypus	8	0
Chicken	26	1
Lizard	7	0
Frog	19	1
Zebrafish	4	1
Sum	**203**	**3**

Table 2: Number of individual nets where the number of aligning 1:1 orthologs increases or decreases after applying chainCleaner.

3.4 chainCleaner keeps aligning blocks in high-scoring nets

As shown in Figure 1C, true CBAs can break a chain into individual nets and some of these nets are filtered out because their score is below a minimum threshold. We expected chainCleaner to produce longer nets,

which should result in more aligning bases in nets above a score threshold. To test this, we determined how many bases overlap aligning blocks of high-scoring nets, before and after applying chainCleaner. As shown in Table 3A, chainCleaner adds between 111 kb (zebrafish) and 1.3 Mb (opossum) in aligning blocks in nets exceeding a score threshold of 100,000. Consistent results were found when we applied a score threshold of 200,000 (Table 3B) or the UCSC "syntenic net" thresholds (Kent, et al., 2003) (Table 3C, Supplementary Figure 18). This shows that chainCleaner improves pairwise genome alignments by removing true CBAs, which in turn leads to additional alignments that pass the net score filter and thus a higher sensitivity.

A

	kb in aligning blocks in nets scoring > 100000		
Query species	**after chainCleaner**	**before chainCleaner**	**difference**
Horse	1,720,962	1,720,300	662
Cow	1,446,590	1,445,969	621
Mouse	1,033,929	1,033,560	369
Opossum	381,810	380,496	1,314
Platypus	162,353	161,778	574
Chicken	116,118	115,653	466
Lizard	83,863	83,746	117
Frog	54,822	54,575	247
Zebrafish	29,507	29,395	111

B

	kb in aligning blocks in nets scoring > 200000		
Query species	**after chainCleaner**	**before chainCleaner**	**difference**
Horse	1,703,626	1,702,679	947
Cow	1,429,550	1,428,465	1,085
Mouse	1,022,979	1,022,485	495
Opossum	371,890	370,534	1,356
Platypus	146,770	146,432	338
Chicken	110,775	110,487	288
Lizard	77,354	77,273	80
Frog	47,270	47,123	147
Zebrafish	17,941	17,919	22

C

	kb in aligning blocks in syntenic nets		
Query species	**after chainCleaner**	**before chainCleaner**	**difference**
Horse	1,710,241	1,709,757	484
Cow	1,442,880	1,442,151	729
Mouse	1,030,775	1,030,272	504

Table 3: chainCleaner keeps aligning blocks in nets above a minimum score threshold.

(A) Score threshold of 100,000. (B) Score threshold of 200,000. (C) "Syntenic nets" are defined in (Kent, et al., 2003) as either nets with a high score (>300,000 for top-level nets) or they are nested in such a high-scoring net and align to the same genomic locus. This syntenic net filter is usually applied to alignments of well-assembled placental mammal genomes.

4 Discussion

Pairwise alignment chains and nets are a widely used concept for genome alignments. While chains represent both paralogous and orthologous alignments, nets attempt to capture only orthologous alignments by taking the entire top-scoring chain for a given genomic locus and filling in un-aligning regions with parts of nested chains. The hierarchy of nets should ideally represent the genome rearrangement history. However, as we have shown here, chain-breaking alignments in top-scoring chains can break nested chains into smaller individual nets and result in a net structure that does not represent the correct rearrangement history.

Here, we developed chainCleaner to detect and remove such CBAs. This helps to correctly infer genomic rearrangements from nets, for example by avoiding an inflation of the number of rearrangements that actually occurred (Figure 1). Furthermore, chainCleaner can help to correctly infer nested rearrangements such as a smaller inversion that happened within a larger inverted region. In such a case, the alignments of the nested (second) inversion are co-linear with the not-inverted flanking alignments and thus are part of the top-level chain (Supplementary Figure 21). chainCleaner can remove the alignments of the nested inversion from the top-level chain and adds them back as a new chain that can become the level 3 net. This results in a hierarchical net structure that correctly represents the nested order of these inversion events (Supplementary Figure 21).

Apart from obscuring the rearrangement history, CBAs can break nested chains into smaller individual nets, which can be subsequently filtered out based on their score. As shown here, removing such CBAs adds new alignments in high-scoring nets. Furthermore, CBAs can be alignments between exons of a gene and a processed pseudogene, which can incorrectly suggest the loss of this gene in the query species (Figure 1C). Removing these CBAs exposes the true alignments between exons of these orthologous genes. With its fast runtime, chainCleaner adds little to the computational burden of computing genome alignments and thus has broad applicability to improve the specificity and sensitivity of genome alignments.

Acknowledgements

We are grateful to the UCSC genome browser group for providing software, genomes and genome annotations. We thank the anonymous reviewers and Juliana Roscito for helpful comments on the manuscript, Katrin Sameith for help with visualization and the Computer Service Facilities of the MPI-CBG and MPI-PKS for their support.

Funding

This work has been supported by the Max Planck Society.

Conflict of Interest: none declared.

References

Amores, A., *et al.* Zebrafish hox clusters and vertebrate genome evolution. *Science* 1998;282(5394):1711-1714.

Angiuoli, S.V. and Salzberg, S.L. Mugsy: fast multiple alignment of closely related whole genomes. *Bioinformatics* 2011;27(3):334-342.

Birney, E., *et al.* Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 2007;447(7146):799-816.

Blanchette, M., *et al.* Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res* 2004a;14(12):2412-2423.

Blanchette, M., *et al.* Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res* 2004b;14(4):708-715.

Bray, N. and Pachter, L. MAVID: constrained ancestral alignment of multiple sequences. *Genome Res* 2004;14(4):693-699.

Brudno, M., *et al.* Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics* 2003a;4:66.

Brudno, M., *et al.* LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res* 2003b;13(4):721-731.

Clarke, S.L., *et al.* Human developmental enhancers conserved between deuterostomes and protostomes. *PLoS Genet* 2012;8(8):e1002852.

Cooper, G.M., *et al.* Distribution and intensity of constraint in mammalian genomic sequence. *Genome Res* 2005;15(7):901-913.

Darling, A.E., Mau, B. and Perna, N.T. progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PloS one* 2010;5(6):e11147.

Dewey, C.N. Whole-genome alignment. *Methods in molecular biology* 2012;855:237-257.

Dubchak, I., *et al.* Multiple whole-genome alignments without a reference organism. *Genome Res* 2009;19(4):682-689.

Earl, D., *et al.* Alignathon: a competitive assessment of whole-genome alignment methods. *Genome Res* 2014;24(12):2077-2089.

Frith, M.C. and Kawaguchi, R. Split-alignment of genomes finds orthologies more accurately. *Genome Biol* 2015;16:106.

Grabherr, M.G., *et al.* Genome-wide synteny through highly sensitive sequence alignment: Satsuma. *Bioinformatics* 2010;26(9):1145-1151.

Harris, R.S. Ph.D. Thesis: The Pennsylvania State University; 2007. Improved pairwise alignment of genomic DNA.

Herrero, J., *et al.* Ensembl comparative genomics resources. *Database : the journal of biological databases and curation* 2016;2016.

Hiller, M., *et al.* Computational methods to detect conserved non-genic elements in phylogenetically isolated genomes: application to zebrafish. *Nucleic Acids Res* 2013;41(15):e151.

Hiller, M., *et al.* A "forward genomics" approach links genotype to phenotype using independent phenotypic losses among related species. *Cell Rep* 2012;2(4):817-823.

Hillier, L.W., *et al.* Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature* 2004;432(7018):695-716.

Kent, W.J., *et al.* Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences of the United States of America* 2003;100(20):11484-11489.

Kinsella, R.J., *et al.* Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database : the journal of biological databases and curation* 2011;2011:bar030.

Lin, M.F., *et al.* Locating protein-coding sequences under selection for additional, overlapping functions in 29 mammalian genomes. *Genome Res* 2011;21(11):1916-1928.

Lindblad-Toh, K., *et al.* A high-resolution map of human evolutionary constraint using 29 mammals. *Nature* 2011;478(7370):476-482.

Lowe, C.B., Bejerano, G. and Haussler, D. Thousands of human mobile element fragments undergo strong purifying selection near developmental genes. *Proceedings of the National Academy of Sciences of the United States of America* 2007;104(19):8005-8010.

Ma, J., *et al.* Reconstructing contiguous regions of an ancestral genome. *Genome Res* 2006;16(12):1557-1565.

McLean, C.Y., *et al.* Human-specific loss of regulatory DNA and the evolution of human-specific traits. *Nature* 2011;471(7337):216-219.

Paten, B., *et al.* Cactus graphs for genome comparisons. *Journal of computational biology : a journal of computational molecular cell biology* 2011a;18(3):469-481.

Paten, B., *et al.* Cactus: Algorithms for genome multiple sequence alignment. *Genome Res* 2011b;21(9):1512-1528.

Paten, B., *et al.* Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res* 2008;18(11):1814-1828.

Pollard, K.S., *et al.* An RNA gene expressed during cortical development evolved rapidly in humans. *Nature* 2006;443(7108):167-172.

Prabhakar, S., *et al.* Accelerated evolution of conserved noncoding sequences in humans. *Science* 2006;314(5800):786.

Prudent, X., *et al.* Controlling for Phylogenetic Relatedness and Evolutionary Rates Improves the Discovery of Associations Between Species' Phenotypic and Genomic Differences. *Molecular biology and evolution* 2016;33(8):2135-2150.

Robin, X., *et al.* pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* 2011;12:77.

Rosenbloom, K.R., *et al.* The UCSC Genome Browser database: 2015 update. *Nucleic Acids Res* 2015;43(Database issue):D670-681.

Siepel, A., *et al.* Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res* 2005;15(8):1034-1050.

Stark, A., *et al.* Discovery of functional elements in 12 Drosophila genomes using evolutionary signatures. *Nature* 2007;450(7167):219-232.

Waterston, R.H., *et al.* Initial sequencing and comparative analysis of the mouse genome. *Nature* 2002;420(6915):520-562.