論文 / 著書情報
Article / Book Information

題目(和文)	Hi-C法を活用した染色体レベルのハプロタイプゲノム構築手法の開発
Title(English)	Development of a chromosome-level haplotype-resolved genome assembly tool using Hi-C
著者(和文)	大内俊
Author(English)	Shun Ouchi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12745号, 授与年月日:2024年3月26日, 学位の種別:課程博士, 審査員:伊藤 武彦,本郷 裕一,立花 和則,二階堂 雅人,山田 拓司
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12745号, Conferred date:2024/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

令和 5 年度　博士論文

Hi-C 法を活用した染色体レベルの ハプロタイプゲノム構築手法の開発

指導教員　伊藤　武彦　教授

東京工業大学
大学院 生命理工学院
生命理工学系　生命理工学コース

大内　俊

第 1 章　序論

1.1 ゲノム配列決定の歴史

ゲノムとはある生き物のすべての遺伝情報のことで、具体的には DNA の４つの塩基、アデニン(A) 、シトシン(C)、グアニン(G)、チミン(T)の配列として遺伝情報は格納されている。ある生き物のゲノム配列を決定することは、その生物の特徴や進化の過程を調べるために重要であるため、様々なシーケンス(ゲノム配列決定)技術が開発され、多種多様な生物種のゲノム配列が決定されてきた。

初期のシーケンス技術としては、1977 年に Maxam-Gilbert 法[1]、Sanger 法[2]が発表されている。どちらも、決定したいゲノム配列の特定の塩基で終わる部分配列を準備し、電気泳動により長さ順に部分配列を並べることで塩基配列を決定するもので、初期のゲノム配列決定に使用された。しかし手動による作業が多く、時間がかかるため、読むことができる塩基配列長は少なかった。その後、Sanger 法は改良され、蛍光検出法[3–5]やキャピラリー電気泳動[6–8]などにより、処理の高速化、並列化、自動化が可能になり、1987 年には Applied Biosystems 社により世界初の自動 DNA シーケンサーABI 370 が発売された。このような技術革新により、1995 年には生物として初めてインフルエンザ桿菌[9]が、1996 年には真核生物として初めて酵母[10]の全ゲノム配列が決定された。その後、1990 年にはヒトの全ゲノム配列の解読を目標としたヒトゲノム計画(Human Genome Project: HGP)[11]がスタートし、2001 年には国際コンソーシアムとセレラ社によりヒトのドラフトゲノム配列(ゲノム全体の約 90%を解読した状態)の論文が発表された[12,13]。両者は、それぞれ別の方法でヒトゲノムを決定しており、前者は階層的ショットガン法、後者は全ゲノムショットガン法を用いている。階層的ショットガン法とは、ゲノムの一部を BAC やフォスミドなどを用いてクローニングし、その配列を決定した後、遺伝地図や物理地図の情報をもとにその配列を並べることでゲノム配列を決定する方法である。一度に決定するゲノム領域を絞ることにより、ミスアセンブルの可能性は低く、得られた配列の精度は高いが、遺伝地図や物理地図が必要なため非常にコストがかかる。一方、全ゲノムショットガン法(Whole Genome Shotgun: WGS)は、全ゲノムをランダムに断片化し、その断片の配列を決定した後に、得られた大量の断片配列(リード)をオーバーラップ情報などをもとに繋ぎ合わせることでゲノム配列を決定する方法である。WGS は階層的ショットガン法に比べコストは安いが、ゲノム配列上の繰り返し配列(リピート配列)などでリードを間違えて繋がないよう高度なアルゴリズムが求められる。シーケンサーのスループットの向上により、大量のリードデータを安価に入手できるようになってきたことや大量のリードデータを扱うための効率的なアルゴリズムの開発が進んだことにより、WGS がゲノム配列決定に使用されるようになっていった。

2000 年代に入り、次世代シーケンサーと呼ばれる Sanger 法とは原理が異なるハイスループットなシーケンサーが登場した[14]。代表的なものとしては、454 社(Roche 社)、Illumina 社製がある。これにより、1 塩基当たりのシーケンスコストが大幅に下がり、大量のシーケンスデータが取得可能になった。これらのシーケンサーが出力するリードは、高精度であるが、リード長は数百 bp と短いため、ショートリードと呼ばれている。また、これらのシーケンサーには、ライブラリを構成する DNA 断片の両端から配列を取得できるという特徴がある。この特徴を利用し、直線状の DNA 断片の両端から配列を決定する Paired-End シーケンスや、あらかじめ環状化された DNA 断片の両端から配列を決定する Mate-Pair シーケンスが開発され、リードのペア間のより長い距離（Paired-End リード：300–500 bp、Mate-Pair：2–30 kb）の情報がゲノム配列決定に使用された。ショートリードは、発表された当初はリード長が短いことからゲノム配列決定へ使用されることは少なかったが、2010 年にジャイアントパンダのゲノム配列決定[15]に使用されその有用性が示されて以降、様々な生物のゲノム配列決定に使用されるようになった。しかし、ショートリードはリード長が短いため、ゲノム上の長いリピート配列を構築するには課題があった。

このようなショートリードの課題を克服するため、ロングリード技術が開発された。代表的なものとして、Pacific Biosciences 社の PacBio リード(continuous long read、CLR)[16]や Oxford Nanopore Technologies 社の ONT リード[17]がある。PacBio リードでは、ポリメラーゼがヌクレオチドを伸長させる際に取り込む蛍光標識したヌクレオチドをセンサーにより読むことで塩基配列を決定する。ONT リードでは、ナノメートルサイズの小さな穴(ポア)を 1 本鎖 DNA 分子が通過する際のイオンの流れのパターンから塩基配列を決定する。これらのロングリード技術は、数十〜数百 kb の長いリードを生成するため、ショートリードでは構築が難しかったリピート配列の問題を低減することができる。しかし、ロングリードにはエラー率が高いといった課題がある。

2019 年には、エラー率が低い高精度なロングリードとして PacBio 社から high fidelity read (HiFi リード)が発表された[18]。同じゲノム領域を繰り返し読み、そのコンセンサスを取ることにより、リード長：約 20 kb、精度：99%以上を達成している。これにより、様々な生物の高品質なゲノム配列決定が可能になった。しかし、HiFi リードを用いてもセントロメア領域などに代表される数百 kb 以上の長いリピート配列を解決することは難しい。

そこで、近年 Hi-C 法[19]という手法がゲノム配列決定に応用されている。Hi-C(High-throughput Chromosome Conformation Capture)法とは染色体の高次構造解析のために開発された手法の一つで、次世代シーケンサーを用いて空間的に近接しているゲノム領域の情報を網羅的に取得することができる(図 1)。Hi-C 法では、まずホルムアルデヒドで細胞内のタンパク質と DNA を架橋させ染色体の立体構造を固定した後、DNA 鎖を制限酵素で切断する。次に切断された DNA 末端をビオチンで標識した後、ライゲーション

を行い空間的に近接している DNA 末端同士を結合させる。最後に DNA を断片化し、ビオチンで標識されているライゲーションされた DNA 断片のみを Paired-End シーケンスで配列を決定する。得られた Hi-C リードのペアは、それぞれライゲーションされた異なるゲノム領域由来の配列であり、空間的に近接しているほどライゲーションされる確率が高くなるため、ゲノム領域間の距離が近いほど、得られる Hi-C リードペアは多くなる傾向がある。そのため、マッピングされる Hi-C リードペア数の大小から、ゲノム領域間の距離情報を得ることができる。Hi-C 法では、Mate-Pair シーケンスなどと比較して数 Mb 離れた遠距離のゲノム間の距離情報を得ることができるため、その情報をゲノム配列決定に使用することで長いリピート配列を解決することができる。実際、Hi-C リードは多くのゲノム配列決定プロジェクトで用いられ、近年では染色体レベルで繋がったゲノムを構築するために利用されている(ヤギ[20]、水牛[21]、蚊[22]など)。

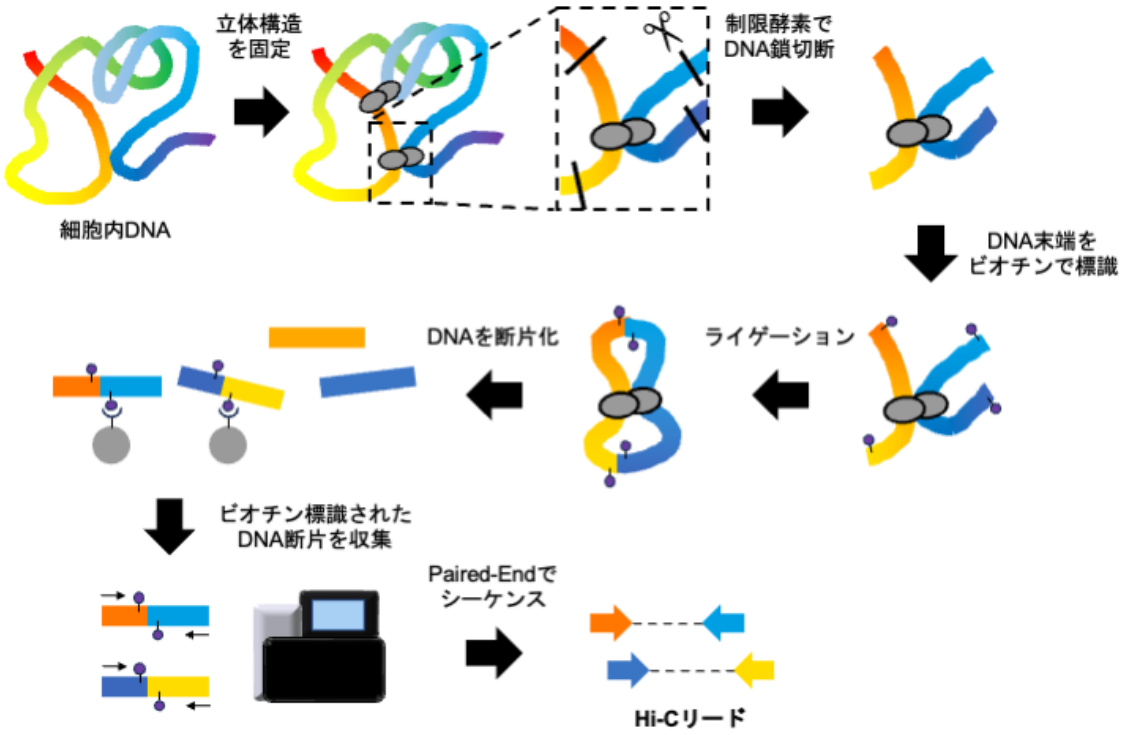

図 1　Hi-C 法の模式図

1.2 ゲノム配列決定の方法

未知のゲノム配列を新規に決定することは、*de novo* assembly(新規ゲノム配列構築)と呼ばれ、一般的に以下の手順で行われる(図 2)。

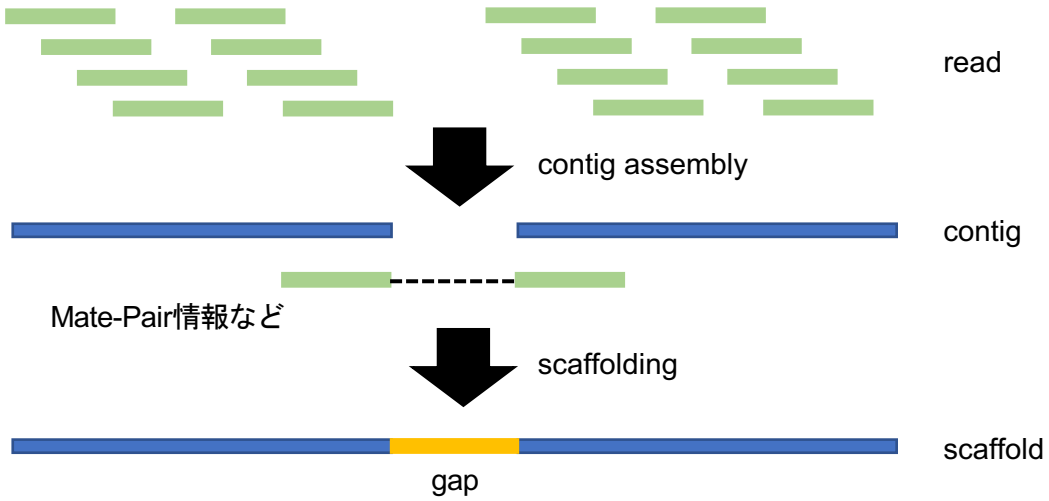

図 2　*de novo* assembly の流れ

はじめに、対象の生物から抽出した DNA をシーケンサーで読み取り、リードと呼ばれる断片を得る。しかし、シーケンサーが一続きで読める配列の長さには制限があるため、リードは断片化してしまう。

そこで、次に contig assembly という過程で、リード間のオーバーラップ情報をもとにリードを繋ぎ合わせ contig と呼ばれる配列を生成する。Contig assembly のアルゴリズムは大きく分けて Overlap-Layout-Consensus(OLC)と de Bruijn Graph(DBG)の二つに分けることができる(図 3)。OLC は、リード間のオーバーラップ情報をもとにリードを伸長させていく方法である。この方法は、高精度で長い Sanger 配列を処理するために開発された手法で CELERA[23]や MIRA[24]といったツールで使用されている。しかし、OLC は全リード間のオーバーラップを計算するため計算コストが高く、大量のリードを処理するのには向いていないため、ハイスループットなショートリードを用いたアセンブリにはあまり用いられない。そこで、DBG が開発された。この方法では、まずリードを *k* 文字の部分文字列(*k*-mer)に分割し、*k*-mer をノード、*k*-1 文字のノード間のオーバラップをエッジとしたグラフを作成する。その後、このグラフの各経路を 1 回だけ通るパス(オイラーパス)を探索することで contig を作成する。オイラーパス探索は効率的なアルゴリズムが知られており、高速に大量のリードを処理することができる。そのため、DBG は多くのショートリード用のアセンブルツールで使用されている(ABySS[25]、VELVET[26]、SOAPdenovo[27]、ALLPATHS-LG[28]、SPAdes[29]など)。しかし、DBG は

k-1 文字の完全一致によりエッジを作成しているため、エラーの多い CLR リードや ONT リードのアセンブリには向いていない。そこでエラーの多いロングリードを用いたアセンブルツールでは、OLC を用いているものが多い(Canu[30]、Flye[31]、FALCON[32]、Wtdbg2[33]など)。また、高精度なロングリードである HiFi リードのアセンブリには、OLC を使用したもの(hifiasm[34]、HiCanu[35]など)、DBG を使用したもの(MBG[36]、LJA[37]など)どちらも開発されている。

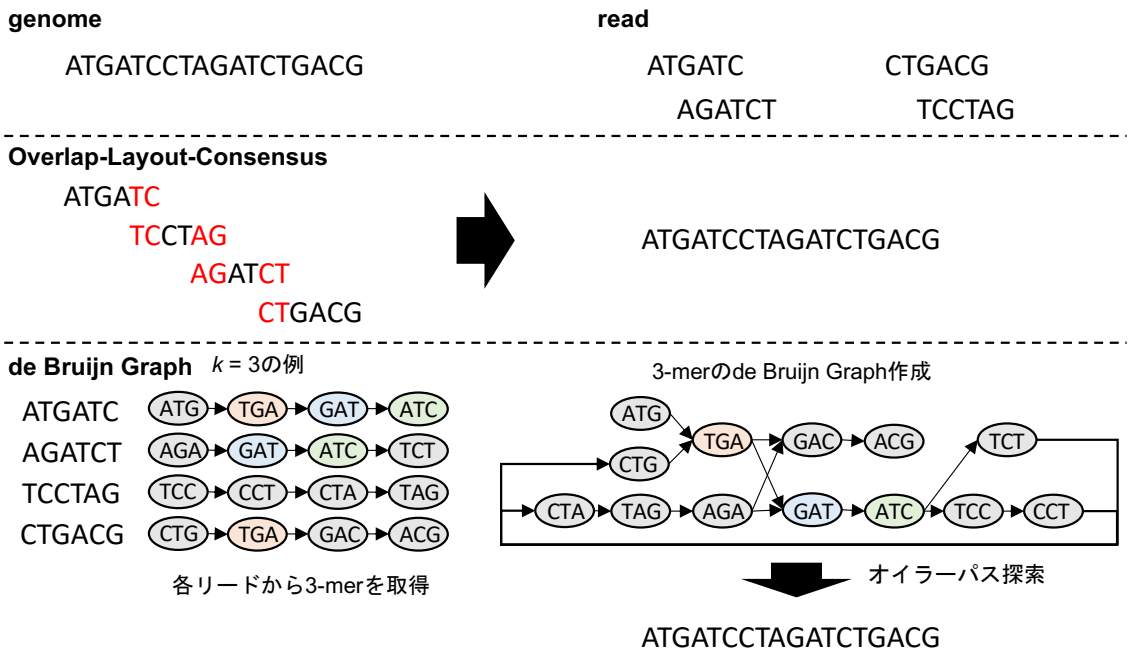

図 3　contig assembly の方法の比較

しかし、リード長より長いリピート領域を解決することは難しいため、contig assembly のみで染色体レベルで繋がったゲノム配列を構築することは難しい。

そこで、次に scaffolding という処理を行い、contig 同士の向きと並びを推定し gap を含めて繋ぐことで scaffold と呼ばれる配列を生成する。Scaffolding では、contig 同士を繋ぐために追加の情報が必要であり、Mate-Pair、Fosmid、BAC、Optical map、Genetic map といった情報が scaffolding に用いられてきた[38]。Scaffolding に使用される情報の特徴を表 1 にまとめる。

Mate-Pair は、ゲノム配列を長い DNA 断片(2–30 kb)に分割し、その両端をシーケンスすることで、距離が既知のリードのペアを得る方法である。この技術は、SSPACE[39]、Platanus[40]などの scaffolding ツールで使用されており、リードペア間の距離情報が scaffolding に使用される。しかし、Mate-Pair ではリードペア間の距離が最大でも 30 kb で、それ以上離れた距離の情報を得ることはできない。

Fosmid は、F-プラスミドの複製起点と分割メカニズムを利用して、長い DNA 断片の

クローニングを可能にした DNA ベクターを用いる方法で、最長 40 kb の長い DNA 配列を得ることができる。BAC は、さらに長い DNA 断片のクローリングを可能にするために開発されたバクテリア人工染色体という DNA ベクターを用いる方法で、最長 200 kb の DNA 配列を得ることが可能である。しかし、Fosmid、BAC ともに作成にはコストがかかるという欠点がある。

Optical map を用いた方法では、制限酵素マップの情報をもとに scaffolding を行う。制限酵素マップとは、制限酵素サイト(制限酵素が認識し切断する特定の塩基配列)のゲノム上の位置を地図にしたもので、長いゲノム DNA をガラス基盤上に固定したのち、制限酵素で DNA を切断、蛍光により切断断片の長さを測定することで作成される。Contig 内の制限酵素サイトの位置と制限酵素マップを照らし合わせることで scaffolding は行われる。Optical map を用いた方法では、長い gap を超えて contig を接続できるが、ゲノムサイズが大きい生物の十分な制限酵素マップ作成は非常にコストが高い。

Genetic map とは、同一染色体上の距離が近い遺伝子ほど、共に遺伝する確率が高いという遺伝子連鎖の考えを基に、染色体上の相対位置を決定して作られた遺伝子の位置を表す地図である。Genetic map は、染色体全体にわたる情報を得ることができるが、構築に高いコストがかかるため、ヒトなどのモデル生物に適用が限定されている。

このように、Mate-Pair、Fosmid、BAC を用いても、200 kb 以上のリピート領域を跨いで contig を接続することは困難であり、セントロメア領域など長いリピート領域を解決することは難しい。また、Optical map、Genetic map は長距離の scaffolding に有効ではあるが、コストが非常に高く、非モデル生物で使用することは難しい。

そこで、近年 Hi-C 法が scaffolding に応用されている。Hi-C 法では、Mate-Pair などとは異なり insert 長(リードペア間の距離)がゲノムの空間的距離に依存して幅広いため、1 kb~数 Mb の幅広い距離の情報を得ることができる。また、Hi-C ライブラリー作製には特殊な機器は不要で、実験コストは比較的低く、非モデル生物でも作成可能である。このように Hi-C 法は低コストで遠距離のゲノム領域間の情報が得られるため、様々な生物のゲノムアセンブル時の scaffolding に適用可能な強力な技術といえる。

表 1　Scaffolding 方法の比較

Scaffolding 方法	距離	コスト	スループット
Mate-Pair	2–30 kb	低	高
Fosmid	40 kb	中	低
BAC	40–200 kb	高	低
Optical map	20–500 kb	高	高
Genetic map	染色体レベル	非現実的	非常に低い
Hi-C	1 kb–20 Mb	低	高

Xu & Dixon, 2019 [38] Table1 より引用し、一部改変

1.3 Hi-C データを活用した既存の scaffolding 手法

Hi-C データを活用した scaffolding では、Hi-C データの以下の二つの特徴を利用している。一つ目の特徴は、Hi-C 接触頻度(Hi-C リードペア数)は、同一染色体内ではるかに多く、染色体間で少ない傾向があることである。これは、各染色体は核内で空間的に異なる領域に配置されており染色体間の空間的距離は遠いため、異なる染色体間がライゲーションされる確率は低くなるからである。この特徴により、マッピングされる Hi-C リードペア数から二つの contig が同じ染色体由来かどうか判定することができる。二つ目の特徴は、同一染色体内で比較した場合、ゲノム配列上の距離が近いほど、Hi-C 接触頻度が多い傾向があることである。これはゲノム配列(1 次元)上の距離が近いほど空間的な距離も近くなり、ライゲーションされる確率が上がるためである。この特徴により、マッピングされる Hi-C リードペア数の大小から、二つの contig 間の距離を推測することができる。Hi-C Scaffolding では、これらの特徴と contig への Hi-C リードのマッピング結果を用いて、contig の最適な順番と向きを決定する。

Hi-C データを活用した scaffolding を行う計算プログラムは様々開発されている。これらのプログラムは大別して二つの方法に分類することができる。1 つ目は、確率モデルを使用した方法である。この方法では、まず contig の順番と向きをランダムに決定し、その contig の順番、向きで、Hi-C のリードが観察される尤度を計算する。そして尤度が最大になるように、contig の順番、向きを変えていくことで scaffolding を行う。DNA triangulation[41]、GRAAL[42]、instaGRAAL[43]などのツールがこの確率モデルを使用した戦略を採用している。２つ目は、グラフベースの方法である。この方法では、まず contig をノード、contig 間を架橋する Hi-C リードをエッジとしてグラフを作成する。そしてグラフ上で最適なパスを探索し、contig の順番、向きを決定することで scaffolding を行う。LACHESIS[44]、3D-DNA[22]、SALSA[45]、SALSA2[46]などのツールでこのグラフベースの戦略が使用されている。ツールによって、グラフの作成方法、最終的なパスの決定方法が異なる。

これらの Hi-C Scaffolding ツールの中でも 3D-DNA、SALSA2 は、Hi-C Scaffolding に一般的に用いられているプログラムであり、以下にそれぞれのアルゴリズムや特徴を示す。

3D-DNA では、まず scaffolding する前に入力 contig 内のミスを Hi-C の接触頻度パターンの変化を用いて検出し、修正を行う。次に、contig ペア間にマッピングされる Hi-C リードペアを集計し、contig をノード、contig 間を架橋する Hi-C リードをエッジとしたグラフを作成する。エッジの重みには、contig 長で割ることで正規化した Hi-C リードペア数を使用している。そして、エッジの重みが最良のノード同士、すなわち架橋する Hi-C リード数が多い contig のペアを繋いでいくことで scaffolding を行う。

SALSA2 でも、3D-DNA と同様にはじめに入力 contig 内のミス修正を行った後、グラ

フの作成、エッジの重みが最良ノード同士の接続を行うことで scaffolding を行う。3D-DNA との違いとしては、SALSA2 では Hi-C リードの physical coverage を用いてミスを検出している点が挙げられる。Physical coverage とは、その領域を跨ぐ Hi-C リードペア数のことで、ミスの箇所ではその両側間を架橋する Hi-C リードペア数は少なくなるため、physical coverage は低下する傾向がある。そこで physical coverage の低下する領域を SALSA2 ではミスとして検出し修正している。また SALSA2 では、エッジの重みの正規化に contig の長さではなく制限酵素サイト数を用いている点も 3D-DNA とは異なる。さらに SALSA2 には、入力に contig assembly 時のグラフの情報を使用でき、scaffolding を補助する機能がある。

3D-DNA や SALSA2 は、多くの生物種のゲノム決定プロジェクトで Hi-C Scaffolding ツールとして使用されており、染色体レベルで繋がったゲノム配列構築に有用であるが、これらの既存ツールの結果には誤って異なる染色体やゲノム領域を繋ぐミスがあり、最終的に手動による修正が必要な場合が多いという欠点がある。そのため、多くのゲノムプロジェクトでは、Hi-C Scaffolding 結果を目視で確認し手動で修正する工程が行われている。実際、3D-DNA を用いている DNA ZOO project (DNA ZOO, https://www.dnazoo.org) でも、最後の工程で Hi-C コンタクトマップを目視で確認し、ミスを修正している。特に短い contig の向きや順番の決定は、mapping される Hi-C リードの数が少ないため困難で、短い contig の向きや順番は間違って出力されることが多い。

1.4 ハプロタイプ phasing

ヒトのような二倍体生物の染色体は、各親から引き継いだ２つの相同な染色体で構成されている。相同染色体のうちどちらか一方の染色体の塩基配列をハプロタイプと呼び、父方、母方由来のハプロタイプを分けてゲノム配列を決定することを phasing と呼ぶ。Phasing された高品質なハプロタイプリファレンスゲノム配列は、農業[47]や医学[48,49]、その他の多くの生物学分野において重要なリソースであり[50]、全染色体の完全長のハプロタイプのセットを構築することはゲノムアセンブリにおける最終目標と言える。

近交系のモデル生物を対象とした初期のゲノムプロジェクトでは、ハプロタイプ間の違いが少ないサンプルを用いていたため、ハプロタイプ間の違いを考慮する必要性は低かった。その後、野生個体などハプロタイプ間の違いが大きいサンプルのゲノム配列決定も行われるようになったが、技術的な制約からハプロタイプ間の違いを無視し、父方由来と母方由来の配列をモザイク状に繋ぎ合わせた擬似的な”一倍体”ゲノム配列(コンセンサス配列)が構築されていた。しかし、近年ハプロタイプ間で違いが大きいゲノム領域が表現型に関連する事例が報告されており、ハプロタイプ間の違いを考慮し、それぞれのハプロタイプを分けて構築することが重要であることがわかってきている。

そこで、父方、母方由来のハプロタイプを分けてゲノム配列を構築する phasing が行

われるようになり、様々な phasing 手法が開発された。既存の phasing 手法は統計ベース手法と実験ベース手法に大きく分けることができる。

統計ベース手法では、集団遺伝情報セットを用いて phasing を行う(図 4)。この手法では、連鎖不平衡（集団内で複数の遺伝子座の変異の特定の組合せの頻度が高くなる現象）を利用して、ハプロタイプを決定していく。この手法は局所的には非常に正確にハプロタイプを決定できるが、組換えホットスポットでスイッチエラー（ハプロタイプが切り替わるエラー）が多い、個体特有の変異の phasing ができないといった欠点がある。また、大規模な集団遺伝情報セットが必要で、ヒトなどのモデル生物でしか使用できない。

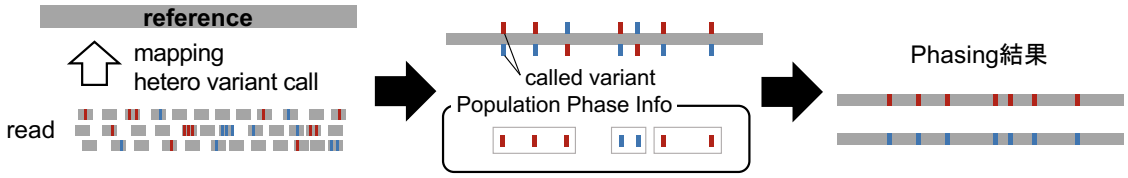

図 4　統計ベース手法

実験ベース手法では、シーケンサーにより得られたリード情報を基に phasing を行う。この手法は、統計ベース手法では phasing できない個体特有の変異も phasing することができるという利点がある。実験ベースの手法としては、トリオデータを使用する方法、アライメントベースの方法とアセンブリベースの方法などがある。以下に各方法の概要と特徴を述べる。

・トリオデータを使用する方法

両親のデータを用いて、父親固有、母親固有な *k*-mer を同定し、ハプロタイプ固有のマーカーとして使用することでハプロタイプを構築する。TrioCanu[51]では、ロングリードを親固有の *k*-mer を用いて二つの独立したハプロタイプに分割し、２つのリードセットを独立にアセンブルすることでハプロタイプを構築する(図 5)。Hifiasm trio mode[34]では、アセンブルグラフから片親ハプロタイプ由来のリードを除去することで phasing された contig を作成する。トリオデータ(子供と両親のデータ)を使用する方法は、非常に精度の高いハプロタイプを構築できるが、両親のデータが必要であるため、野生個体など親のデータが手に入りにくいサンプルには使用できないという制限がある。

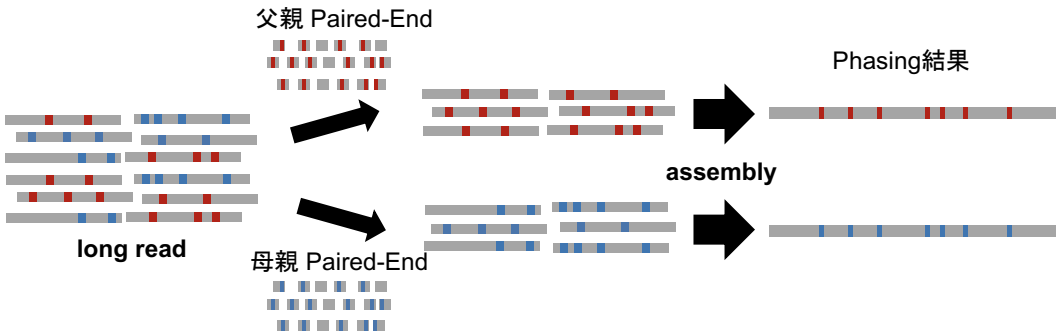

図 5　トリオデータを使用する方法

・アライメントベースの方法

アセンブリに用いたリードをリファレンスゲノムまたは構築したコンセンサス配列にアライメントし、ハプロタイプ間の変異をコール、その後隣接する変異をペアリングすることでハプロタイプを決定する。ショートリードはリード長が短く複数の変異をカバーすることが少ないため、ペアリングできる変異が限られる。そこで、Mate-Pair、ロングリード、Linked read が phasing に用いられている。しかし、これらの手法を用いても、長いホモ領域やセントロメアを超えた phasing をすることは困難である。

そのため、近年では Hi-C 法を phasing に用いるプログラムも登場している(HapCUT[52]、HapCUT2[53]など) 。Hi-C リードはリードペアが染色体全長に及ぶので、この問題を克服することができる。重要なことに、Hi-C リードペアは、同じハプロタイプ間を架橋している可能性が高く、phasing に使えることが示されている[54]。

しかし、アライメントベース手法は、高品質なリファレンス配列が必要で、マッピングベースの変異のコールに依存しているため、精度が悪いことが問題点として挙げられる。特に、ヘテロ接合度が高い(ハプロタイプ間の違いが大きい)領域や構造変異は、リードのマッピングにより変異をコールすることは難しく、ハプロタイプ構築に失敗することがある。

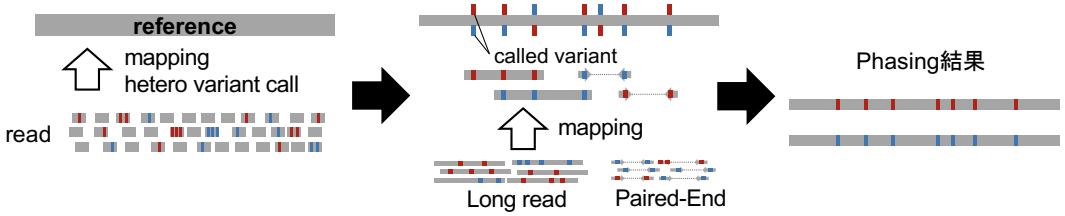

図 6　アライメントベースの方法

・アセンブリベースの方法

ハプロタイプを区別した *de novo* assembly を行うことで、phasing を行う。アライメントベースの方法が、小さい変異のみを対象にしているのに対し、アセンブリベースの方法は、大規模な挿入や逆位など大きい構造変異も phasing することができるという利点がある。

二倍体ゲノムのアセンブルグラフでは、ハプロタイプ間で変異のある領域によるバブルと呼ばれる枝分かれ構造ができる。アセンブリベースの方法では、バブル間を架橋するロングリードや Mate-Pair の情報から、バブル構造を解決することで phasing を行う (Platanus-allee[55]、FALCON-Unzip[32]など)。しかし、ロングリードや Mate-Pair では、数 Mb 以上の長いホモ領域を超えてバブル構造を解決することはできない。

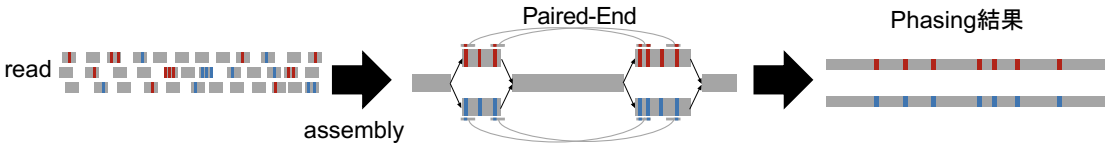

図 7　アセンブリベースの方法

上記のようにこれまで様々な phasing 手法が開発されてきたが、それぞれの手法には課題や制限があった。統計ベース手法は集団遺伝情報のあるモデル生物でしか使用できず、個体特有の変異を phasing できない。トリオデータを用いる方法は、両親データが必要で使用できるサンプルが制限される。アライメントベースの方法は高品質なリファレンスゲノムが必要で構造変異の phasing は困難である。アセンブリベースの方法は、構造変異も phasing することができるが、ロングリードや Mate-Pair では長いホモ領域を超えて phasing をすることはできない。

それゆえ、Hi-C データを用いたアセンブリベースの phasing 手法が開発されている（FALCON-Phase[56]、ALLHiC[57]、hifiasm Hi-C mode[58]）。Hi-C データでは、同じハプロタイプ間を架橋する Hi-C リードペアが多い傾向にあり、リードペア間の距離は最大数 Mb と長距離の情報を得ることができるため、Hi-C データをアセンブリベースの phasing 手法に応用することで長いホモ領域を超えた染色体レベルのハプロタイプを構築できることが期待される。

1.5 Hi-C データを活用した既存の phasing 手法

Hi-C データを活用した既存のアセンブリベースの phasing 手法のアルゴリズム、特徴を以下に述べる。

FALCON-Phase は、FALCON-Unzip の結果を入力に、Hi-C データ用いて phasing を行うツールで、同じハプロタイプを架橋する Hi-C リード数が多くなるように、バブル構

造を解決していくことでハプロタイプを決定する。FALCON-Phase には、Hi-C データによる scaffolding 機能はないが、他の Hi-C Scaffolding ツールと組み合わせることで染色体レベルのハプロタイプを構築できる。しかし、FALCON-Phase で作成された染色体レベルのハプロタイプアセンブリは精度が悪く、大きいスイッチエラー(scaffold の途中でハプロタイプが切り替わるエラー)が多いことが報告されている[59]。

ALLHiC は、多倍体ゲノム用に開発された Hi-C Scaffolding、Phasing ツールである。はじめに、染色体レベルにつながった近縁種ゲノムの遺伝子情報を利用して、異なるハプロタイプ間の Hi-C リンクを除去する。続いて階層的クラスタリングにより、染色体ごとに contig をわけ、最後に遺伝的アルゴリズムにより contig の順番、向きを決定する。ALLHiC は、異なるハプロタイプ間の Hi-C のリンクの除去することで、正確な scaffolding、phasing を可能にしている。しかし、染色体レベルにつながった近縁種ゲノムの遺伝子情報や染色体の本数の情報が必要という欠点がある。

Hifiasm Hi-C mode は、HiFi 用のアセンブラである hifiasm に Hi-C データ用の機能を組み込んだもので、HiFi リードを基に作成したアセンブルグラフを Hi-C データにより phasing を行う。しかし、FALCON-Phase 同様、Hi-C データによる scaffolding 機能はなく、染色体レベルで繋がったハプロタイプを構築することは難しい。

このように、Hi-C データを活用した既存のアセンブリベースの phasing 手法は、近縁種の染色体レベルアセンブリが必要であったり、Hi-C データによる scaffolding 機能がなかったりするため、染色体レベルのハプロタイプ配列を高精度に構築できるツールは乏しいのが現状である。

1.6 本研究の目的

以上、前節までで述べたように、Hi-C 法は scaffolding や phasing に応用されており、染色体レベルのハプロタイプゲノム配列の構築に使われている事例も存在する。しかし、既存の Hi-C Scaffolding ツールは、最終的にマニュアルによる修正が必要な場合が多く、短い contig の向き、順序のミスが多いなどの課題がある。また、既存のアセンブリベースの Hi-C Phasing 手法は、近縁種の染色体レベルアセンブリが必要であったり、Hi-C Scaffolding 機能がなかったりと制限があり、染色体レベルのハプロタイプ配列を高精度に構築できるツールは乏しい。

そこで、Hi-C データを用いて染色体レベルのハプロタイプ配列を構築できるツールの開発を目的として本研究を実施した。Hi-C データを用いた scaffolding、phasing ツール GreenHill を新規に開発し、様々な生物種のデータを用いたベンチマークでその有用性を確かめた。

1.7 本論文の構成

本論文は、Hi-C 法を用いて染色体レベルのハプロタイプ配列を構築できるツールの開発を目的としており、第２章にて開発した GreenHill のアルゴリズムについて詳細に説明する。第３章では、以下に示す様々なデータに対して GreenHill を適用することでその有用性を示す。

・線虫のシミュレーションデータ

線虫(*Caenorhabditis elegans*)の CLR、Hi-C シミュレーションデータに GreenHill を適用し、結果をリファレンス配列と比較することで、詳細な精度の評価を行った。GreenHill が既存ツールと比較して連続性、精度の両方で優れた結果が得られることを確認した。

・ショウジョウバエデータ

ショウジョウバエ(*Drosophila melanogaster*)の HiFi 実データ、Hi-C シミュレーションデータを用いて評価を行い、GreenHill が HiFi データに対しても優れた性能を発揮することを確認した。

・ウシデータ

ウシ(*Bos indicus*×*Bos taurus*)の実データに対してベンチマークを行い、ゲノムサイズが大きいゲノムの実データに対しても GreenHill が既存ツールを上回る性能を発揮することを示した。

・キンカチョウデータ

ヘテロ接合度の高いキンカチョウ(*Taeniopygia guttata*)の実データに GreenHill を適用し、ヘテロ接合度が高いサンプルでも、染色体レベルのハプロタイプ配列を構築できることを示した。

・他生物種データ

セキセイインコ(*Melopsittacus undulatus*)、クロサイ(*Diceros bicornis*)、コチョウザメ(*Acipenser ruthenus*)の実データでベンチマークを行い、GreenHill が生物種に対して高い汎用性を持つことを確かめた。

最後に第４章で以上の総括として全体のまとめと展望を述べる。

第 2 章　Hi-C Scaffolding、Phasing 手法 GreenHill の開発

本研究では、Hi-C データを用いて染色体レベルのハプロタイプゲノムを構築することを目的とし、Hi-C Scaffolding、Phasing ツール GreenHill を新規に開発した[60]。このツールは、様々なアセンブラの結果と Hi-C データを入力として、染色体レベルのハプロタイプゲノムを両親データなしで構築することができるよう設計されている。特徴的な機能として、Hi-C のコンタクトマップを利用した独自のミス修正機能や、Hi-C データ以外のリードデータも同時に用いて精度を高める機能が実装されている。ソースコードについては、筆者が C++言語で実装し、オープンソースとして Github リポジトリ(https://github.com/ShunOuchi/GreenHill) 、 Zenodo データリポジトリ(https://doi.org/10.5281/zenodo.8041374)にて公開している。なお、以後の節で説明するアルゴリズムはバージョン 1.1.0 のものである。

2.1 GreenHill のアルゴリズムの概要

GreenHill のアルゴリズムの全体像を図 8 に示す。GreenHill は、他のアセンブラから構築された contig に加えて、Hi-C およびロングリードを入力として受け取り、scaffolding、phasing を行う。入力の contig は、(i) Paired-haplotype style、(ii) Pseudo-haplotype style、(iii) Haplotype-ignorant style など、どのような形式の contig でも用いることができるようになっている(図 8 a)。GreenHill では、はじめに入力 contig から、相同染色体の同一 Locus 由来の contig を対応づけ、各ペアを一つの”consensus contig”としてマージする(図 8 b)。次に、consensus contig をロングリードや Hi-C を用いて scaffolding を行い、染色体レベルで繋がった consensus 状態の配列を構築する(図 8 c, d)。最後に、ロングリード、Hi-C を用いて phasing を行い、二つのハプロタイプを構築する(図 8 e)。

次節以降で、これらのアルゴリズムの詳細を述べる。

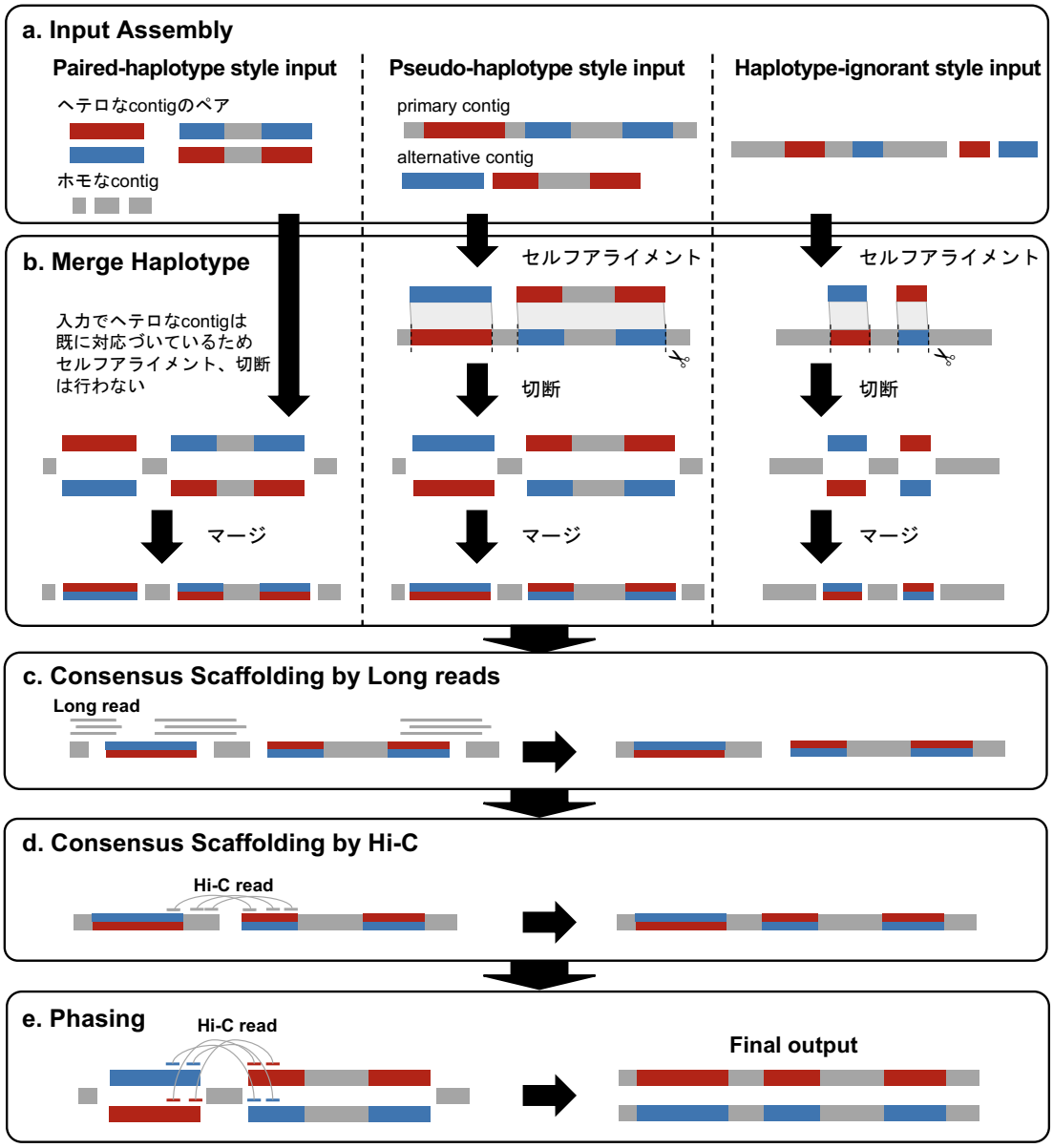

図 8　GreenHill の全体像

2.2 Input assembly

GreenHill では、他のアセンブラで構築された contig を入力として用いており、以下のすべての形式の contig を入力として用いることができる(図 8 a)。

i. Paired-haplotype style

この形式では、ハプロタイプ間で差異が大きいヘテロな領域の contig とハプロタイプ間で差異が少ないホモな領域の contig が分けて出力される。ヘテロな contig は、相

同染色体間で同一 Locus 由来のヘテロな contig と bubble として対応づけられ、塩基配列長が長い方を”primary bubble”、短い方を”secondary bubble”として出力される。Platanus-allee[55]などの結果がこの形式に当たる。

ii. Pseudo-haplotype style

この形式では、primary contig と alternative contig の 2 種類の contig が出力される。Primary contig とは、ヘテロな領域とホモな領域をスイッチエラー(ハプロタイプが切り替わるエラー)を許容しながら可能な限り長く繋げた contig である。一方、alternative contig とは、primary contig に含まれなかった残りのヘテロな領域の contig である。一般的に、primary contig と alternative contig の対応関係の情報は残念ながら出力されない。FALCON-Unzip[32]などの結果がこの形式に当たる。

iii. Haplotype-ignorant style

ハプロタイプに関する情報を出力しない形式を指す。Canu[30]、Flye[31]など多くのアセンブラの結果がこの形式に当たる。これらのツールは、ハプロタイプ間の違いを無視した consensus 状態の配列の構築を前提としているが、ヘテロ接合度の高いサンプルや実行時のオプションによっては、ハプロタイプ間の違いを考慮してヘテロな領域の一部をハプロタイプ別に出力し、推定ゲノムサイズの 1.x〜2 倍のゲノムを構築することがある。GreenHill は、このような部分的にハプロタイプ別で構築されているアセンブル結果も入力として用いることができる。

2.3 Merge haplotype

Merge haplotype では、入力 contig 中のヘテロな領域を検出し、マージを行い consensus 状態の配列”consensus contig”を作成する。

入力の形式が、Pseudo-haplotype style または Haplotype-ignorant style の場合、ヘテロな contig 同士は対立するアリルとして対応づいていないため、はじめにセルフアライメントを行うことで対応づけを行い、入力の形式を Paired-haplotype style に変換する(図 9 a)。以下に詳細な手順を述べる。

i. 入力 contig のセルフアライメントを計算する。ツールは minimap2[61]、オプションは -D –secondary=no -c を用いて入力 contig 同士のアライメントを計算し、Identity が 80%未満のアライメント結果は除去する。

ii. 各 contig について、最も長くアライメントが取れる contig を”opposite contig”とし計算する。

iii. アライメントが長い順に、contig を opposite contig と対応づける。ただし、contig u の opposite contig v が既に別の contig w と対応づいており、u と v 間のアライメントと w と v 間のアライメントが重複している場合、contig u は冗長な配列と判断し除去される。

iv. iiiで対応づけた情報をもとに入力 contig を切断し、contig をホモな領域とヘテロな領域に分ける。

v. 各 contig のハプロタイプの情報と coverage 情報をヘッダーに保存する。対応づけられたヘテロな contig のペアは bubble として保存され、対応つかなかった contig はホモな contig として保存される。各 contig の coverage は、ヘテロな contig はC、ホモな contig は$C \times 2$として保存される(閾値C：デフォルトでは 40)。

入力形式が、Paired-haplotype style の場合、以下の手順で、ヘテロな contig のマージを行う(図 9 b)。

はじめに、入力のヘッダーからハプロタイプ情報と coverage 情報を読み取る。ハプロタイプ情報を用いて、対応する primary bubble と secondary bubble の contig をマージし、consensus contig を作成する。Bubble を形成していない contig (nonBubbleOther)については、coverage が$C_{\mathrm{hetero}} \times r$ 未満の場合はヘテロな contig、$C_{\mathrm{hetero}} \times r$ 以上の場合はホモな contig と分類される。ここで、rは定数で default では 1.75、C_{hetero}はヘテロな contig の平均 coverage で Platanus-allee[55]と同様な方法を用いて計算する。マージ結果は、要素数 2 の配列 T に保存される。Primary bubble contig u と secondary bubble contig v をマージして作成した consensus contig は$[u, v]$として、bubble を形成していないヘテロな contig u は$[u, -]$として、bubble を形成していないホモな contig u は$[u, u]$として保存される。

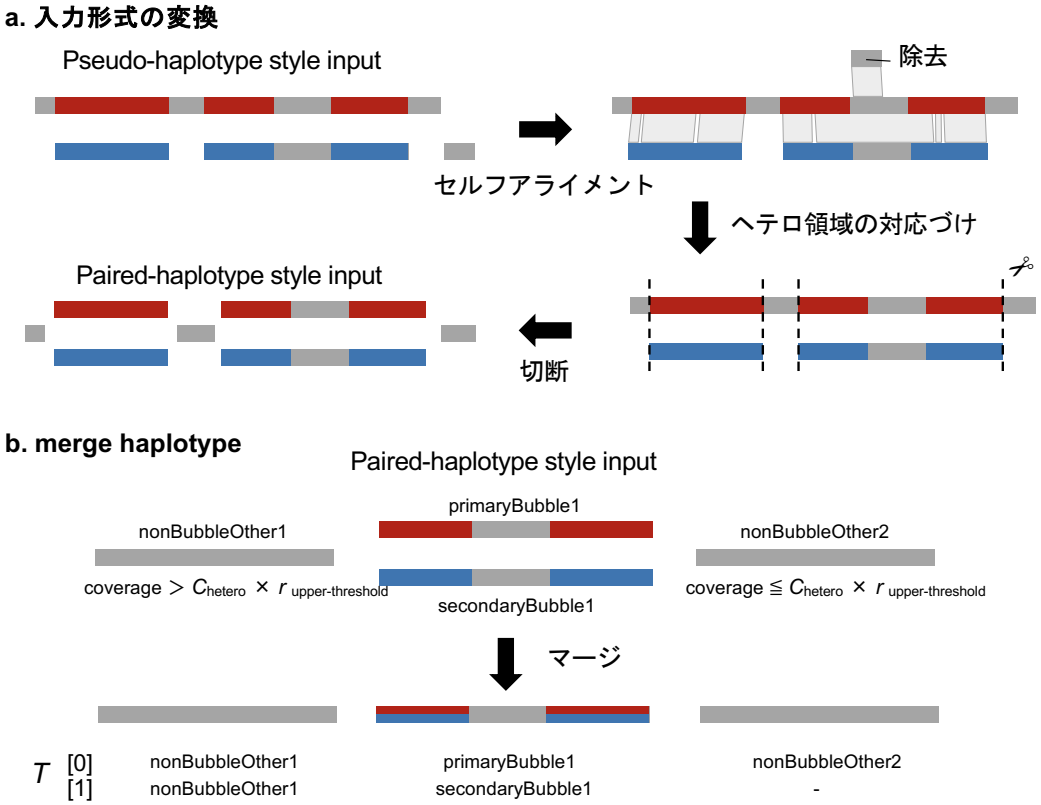

図 9　Merge haplotype

2.4 Consensus scaffolding by long reads

Consensus scaffolding の第一ステップでは、2.3 Merge haplotype でマージした consensus contig を、ロングリード(と Pair-End リード)を用いて scaffolding をする。この処理は、Platanus-allee[55]の scaffolding 処理と類似したアルゴリズムとなっている。GreenHill 独自の機能として、Hi-C を用いたエラーエッジの除去を実装している。ロングリードによる Consensus scaffolding の模式図を図 10 に示す。以下各機能の詳細を述べる。

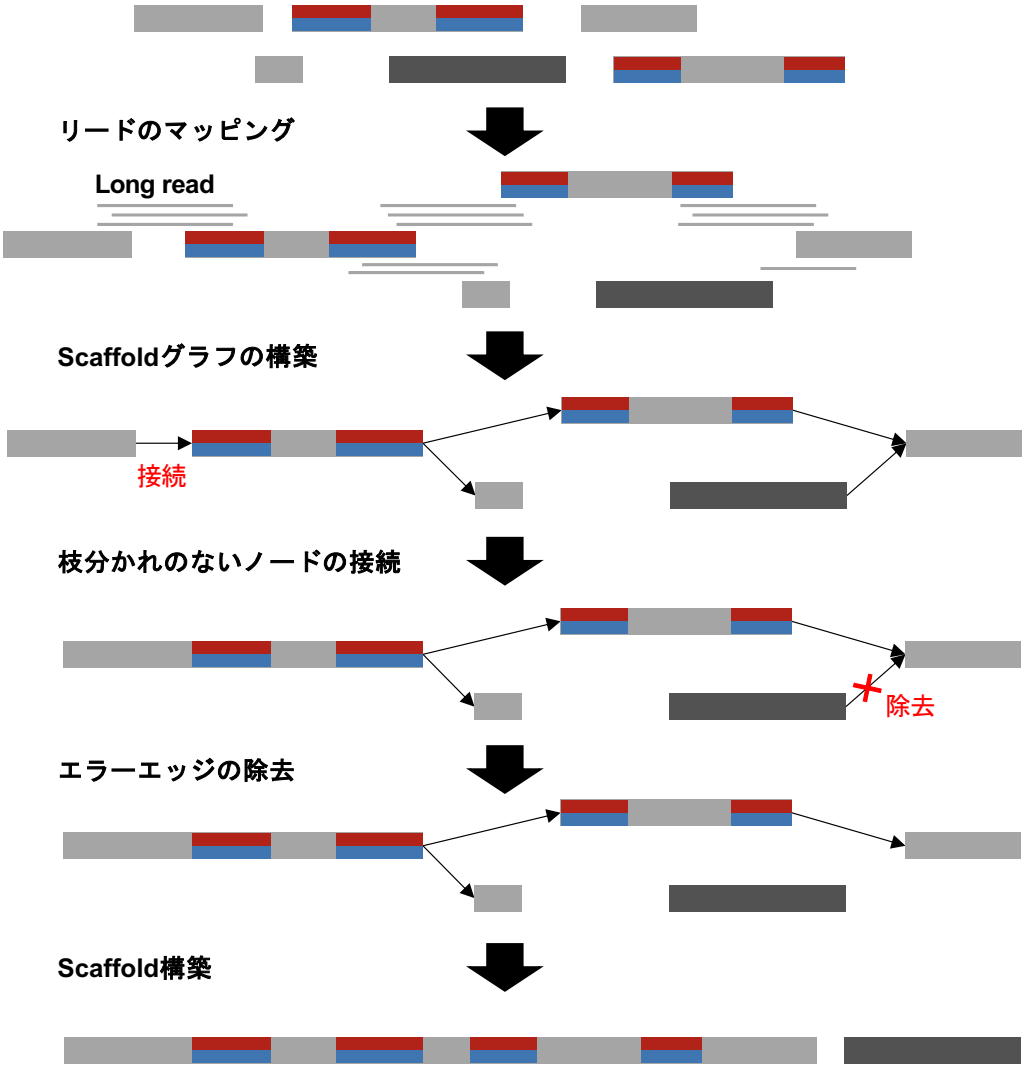

図 10　Consensus scaffolding by long reads

リードのマッピング

ショートリード(Paired-End リードまたは 10X リード)、Hi-C は、*k*-mer を用いてマッピングを行う(図 11)。以下に詳細な手順を述べる。

i. Contig 中の *k*-mer をキーとし、contig 中の位置を値としたハッシュテーブルを作成する。

ii. リード配列から重複なしで、*k*-mer を取り出す。

iii. ユニークな *k*-mer により、マッピング位置を決定する。ここで、ユニークな *k*-mer

とは、全 contig 中に一つだけ含まれる *k*-mer である。

iv. ユニークな *k*-mer がないため、iiiでマッピング位置を決定できなかった場合、bubble ユニークな *k*-mer を用いて、マッピング位置を決定する。ここで bubble ユニークな *k*-mer とは、全 contig 中で出現回数が 2 回であり、同じ bubble を構成する primaryBubble、secondaryBubble にそれぞれ 1 回ずつ含まれている *k*-mer である。この方法では、primaryBubble、secondaryBubble のどちらにマッピングすべきかはわからないため、primaryBubble のみにマッピングし、”consensus link”として保存する。

この方法はユニークな *k*-mer を用いているため、リピート配列によるマッピングミスを防ぐことができる。また、bubble ユニークな *k*-mer を用いることで、対応する primaryBubble、secondaryBubble 間で相同な領域にも、マッピングすることができる。Consensus link は、Consensus scaffolding 時のみに使用し、phasing の際は用いない。

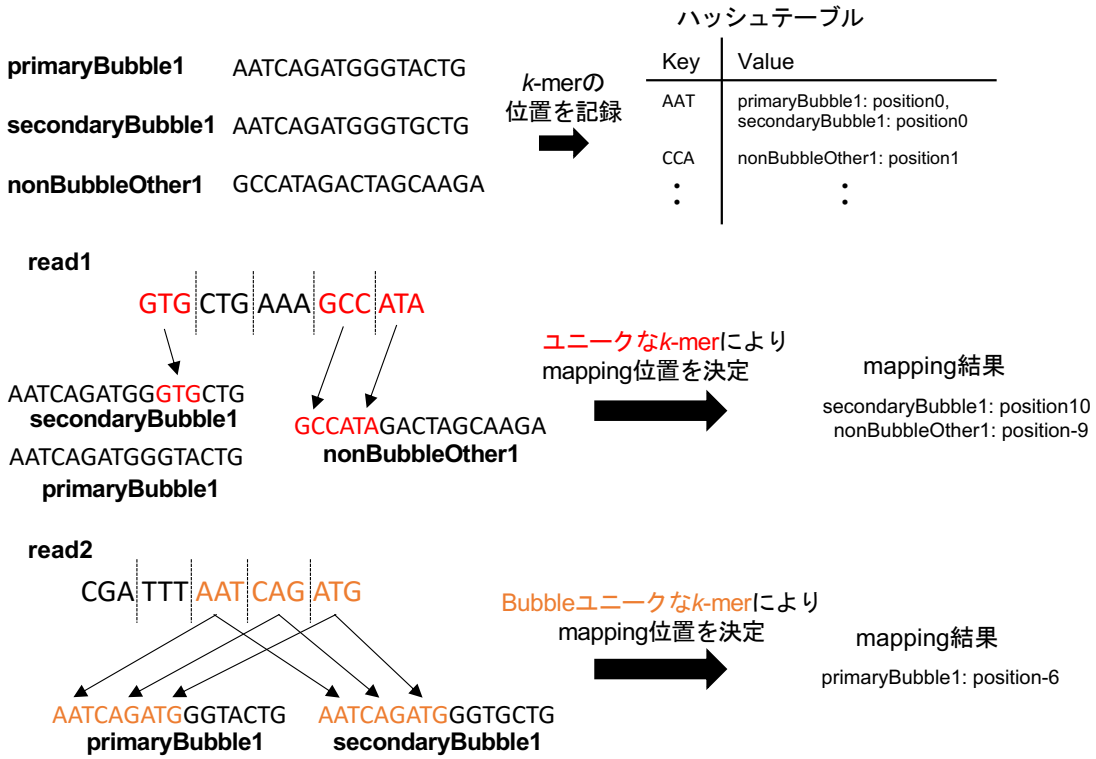

図 11　ショートリードのマッピング

ショートリードのマッピングの模式図。*k*=3 の例。Read1 は、ユニークな *k*-mer により、seecondaryBubble1 と nonBubbleOther1 にマッピングされる。Read2 は、ユニークな *k*-mer がなかったため、bubble ユニークな *k*-mer を用いて primaryBubble1 にマッピングされる。

ロングリードは、以下の手順でマッピングを行う(図 12)。

i. Minimap2 でロングリードを contig にマッピングする。オプションは-c を使用し、塩基レベルのアライメントを計算する。

ii. 重複するアライメントのフィルタリングを行う。複数のアライメント結果でリード上の位置が重なっていた場合、Identity が高い方を残し、低い方を除去する。なお、Identity は塩基の一致数をアライメント長で割ることで計算する。

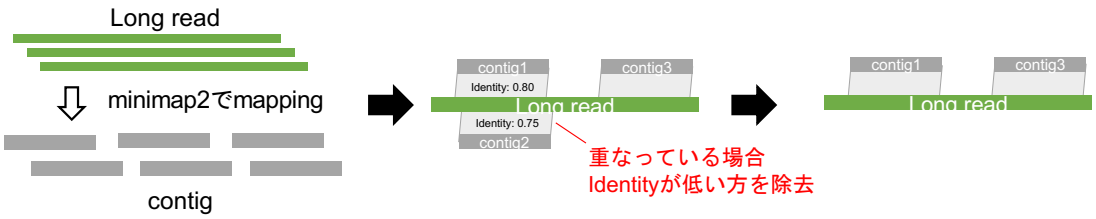

図 12　ロングリードのマッピング

Scaffold グラフの構築

Paired-End リードのペアまたはロングリードが異なる consensus contig にマップされる場合、consensus contig 間にリンクがあるとする。各 consensus contig をノードとして、リンクを閾値 *n* 以上持つ consensus contig 間にエッジを作成し、scaffold グラフを作成する(図 13)。各エッジは、リンク数と consensus contig 間の距離情報を保持する。Consensus contig 間の各リンクから計算した距離の平均をとることで、consensus contig 間の距離は計算する。

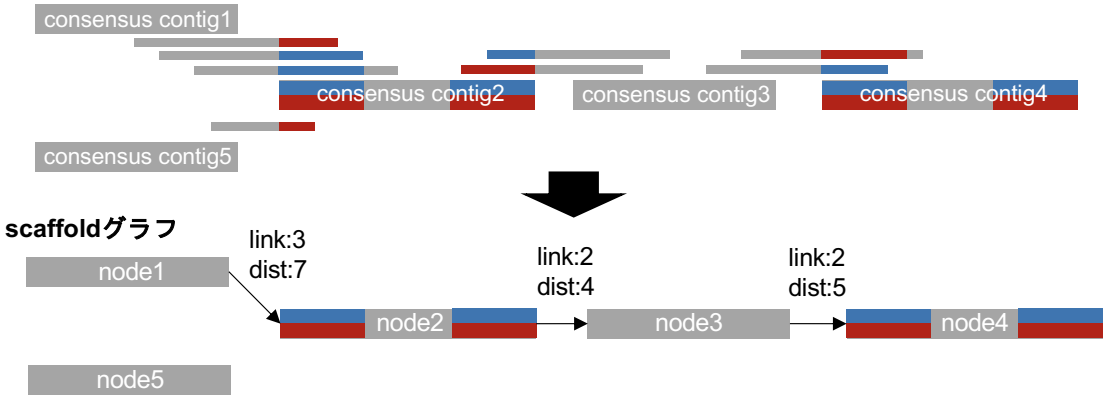

図 13　scaffold グラフの構築

エッジ作成のために必要なリンク数の下限(*n*)が 2 の場合の例。”link”はエッジを支持するリンク数、”dist”は consensus contig 間の距離。

枝分かれのないノードの接続

以下の二つの条件を満たすとき、ノード *u*、*w* を接続する(図 14)。

・ノード *u*、*w* を接続するエッジの link 数 ≧ minLink (default = 3)

・ノード *u*、*w* 間に枝分かれ（他のノードへの link）がない

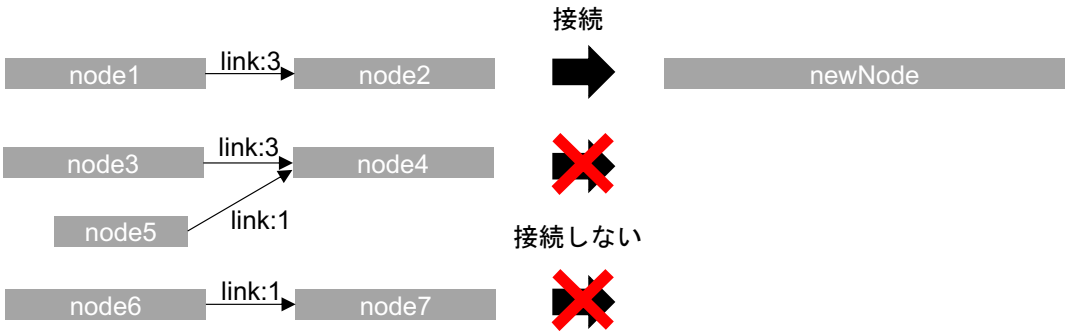

図 14　枝分かれのないノードの接続

node1、node2：ノード間を接続するエッジの link 数≧3 かつ枝分かれなし → 接続

node3、node4：枝分かれあり → 接続しない

node6、node7：接続するエッジの link 数 ＜3 → 接続しない

Hi-C によるエラーエッジの除去

二つのノード間を接続するエッジを Hi-C のコンタクトマップにより正しいか確認し、誤りと判定された場合、そのエッジを除去する。コンタクトマップとは、縦軸横軸ともにゲノム上のポジションとして、交差する点にマッピングされた Hi-C リードペア数を表すヒートマップである。ゲノムをある一定のサイズの bin に区切り、bin 間を架橋する(Hi-C リードのペアがそれぞれの bin にマッピングされる)Hi-C リード数を計算することで、コンタクトマップは作成される。Hi-C 法では二つの特徴（i. 染色体内の Hi-C 接触頻度は、染色体間の Hi-C 接触頻度より高い、ii. 同一染色体内で比較した場合、ゲノム上の距離が近いほど、Hi-C 染色頻度が高い）があるため、Hi-C コンタクトマップでは、対角線に近いほど Hi-C 架橋数が多くなる傾向がある。この性質を利用してエラーエッジを検出し、除去を行う。以下に手順を示す(図 15)。

i. エッジが接続する 2 つのノードを並べ、一定のサイズ(default: 100 kb)に区切り、Hi-C のコンタクトマップを作成する。各ピクセル値$M_{i,j}$は、*i* 番目、*j* 番目の bin を架橋する Hi-C リードペア数に対応する。ノードの長さが 200 kb 未満の場合、短すぎて Hi-C コンタクトマップによるエラーエッジの判定ができないため、エラーエッジの除去は行わない。

ii. 以下の式で定義される分離度 S、R を計算する。

$$S = \sum_k \frac{N_{in,k} N_{out,k} (\mu_{in,k} - \mu_{out,k})^2}{(N_{in,k} + N_{out,k})^2}$$

$$R = \frac{\sum_k L_{out,k}}{\sum_k N_{out,k}}$$

$N_{in,k}$：対角線から k 離れた染色体内のピクセル数
$N_{out,k}$：対角線から k 離れた染色体外のピクセル数
$\mu_{in,k}$：対角線から k 離れた染色体内のピクセルの Hi-C 架橋数の平均
$\mu_{out,k}$：対角線から k 離れた染色体外のピクセルの Hi-C 架橋数の平均
$L_{out,k}$：対角線から k 離れた染色体外で、$\mu_{in,k}$ より Hi-C 架橋数が低いピクセル数
染色体内：コンタクトマップを二つのノードの境界で区切った際の内側の領域
染色体外：コンタクトマップを二つのノードの境界で区切った際の外側の領域

分離度 S は、染色体内、染色体外の領域を二つのクラスとみなした際のクラス間分散を表している。これは大津の二値化法[62]のアイデアを応用しており、染色体内と染色体外間で Hi-C 架橋数に差が大きいほど大きくなるようになっている。分離度 R は、架橋数が平均未満の染色体外のビンの割合を示しており、Hi-C 架橋数が全体的に低いスパースな Hi-C コンタクトマップで、誤ってミスと判定してしまわないように使用している。分離度 S、R の値はどちらも、染色体内と染色体外の Hi-C 架橋数の差が大きいほど大きくなる値で、2 つのノードの境界の上側、下側の両方で計算し、それぞれを S_{upper}、 S_{lower}、R_{upper}、R_{lower} と表記する。

iii. 以下の二つの条件を共に満たすとき、エラーエッジとみなし除去する。

- $\boldsymbol{S_{\mathrm{upper}}/S_{\mathrm{upper},0} + S_{\mathrm{lower}}/S_{\mathrm{lower},0} \geq s\ (\mathrm{constant\ value;\ default} = 1.0)}$
- $\boldsymbol{R_{\mathrm{upper}} + R_{\mathrm{lower}} \geq r\ (\mathrm{constant\ value;\ default} = 1.6)}$

ただし、S_0 は染色体外の架橋を 0 として計算した分離度 S

エラーエッジの場合、ノードの境界を跨ぐ Hi-C リードペア数は少なくなるため、Hi-C コンタクトマップでは染色体外の架橋数が少なくなる。そのため、エラーエッジの場合、染色体内と染色体外の Hi-C 架橋数の差が大きくなると考えられるため、分離度 S と R が閾値を超えた場合にエッジを除去するようにしている。

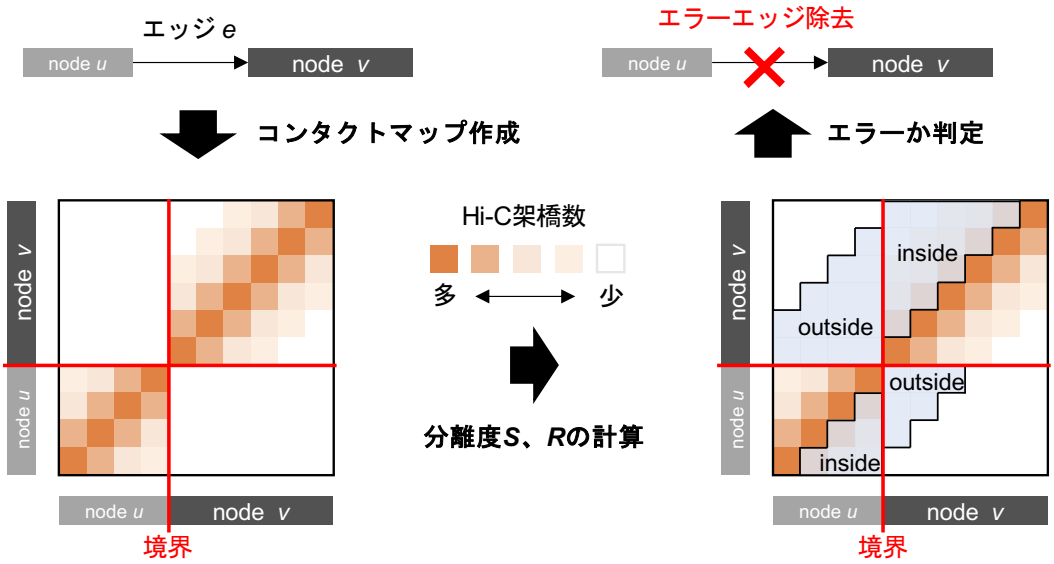

図 15　Hi-C コンタクトマップによるエラーエッジの除去

ノード *u* と *v* を接続するエッジ *e* がエラーかどうか判定する例。コンタクトマップのノードの境界部分に赤線を入れている。コンタクトマップでは、対角線に近いほど Hi-C 架橋数が多い傾向がある。エラーエッジの場合、染色体内(図中の inside)と染色体外(図中の outside)の領域の Hi-C 架橋数の差が大きくなるため、分離度 *S*、*R* の値が大きくなる。分離度 *S*、*R* を用いてエッジ *e* がエラーか判定する。

Scaffold グラフのノードの衝突の判定

Scaffold グラフ中で、ノード *u*、*w* がノード *v* に接続している場合を考える。*v* を基準とした *u* と *w* の位置から、*u* と *w* の位置のオーバーラップを計算し、オーバラップの長さが閾値(default : ロングリードの平均長× 0.05)以上の場合、*u* と *w* は衝突していると判定する(図 16)。

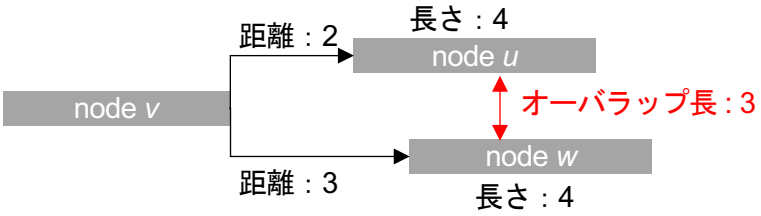

図 16　scaffold グラフのノードの衝突の判定

ノード *v* からのエッジを持つノード *u* と *w* の衝突判定の模式図。ノード *v* を基準とした位置関係からノード *u* と *w* のオーバラップ長は 3 と計算され、オーバラップ長が閾値以上の場合、ノード *u* と *w* は衝突していると判定される。

Scaffold 構築

Platanus-allee と同様の手法で、scaffold を構築する。以下に scaffold 構築の手順を述べる(図 17)。ただし、V_{repeat}を衝突する隣接ノードを持つノードの集合、V_{used}をすでに scaffold 構築に使われたノードの集合とする。

i. $v_0 \notin V_{\mathrm{repeat}}$ and $v_0 \notin V_{\mathrm{used}}$を満たすノード$v_0$をランダムに選び、ノード集合$V_{\mathrm{scaffold}}$を$\{v_0\}$として初期化する。また、$v_0$を$V_{\mathrm{used}}$に追加する。

ii. $u \in V_{\mathrm{scaffold}}$ and $u \notin V_{\mathrm{repeat}}$ and $w \notin V_{\mathrm{scaffold}}$を満たすエッジ$(u, w)$を探し、ノード$w$が$V_{\mathrm{scaffold}}$の要素と衝突しないとき、$w$を$V_{\mathrm{scaffold}}$と$V_{\mathrm{used}}$に追加する。$w$の候補が複数ある場合、$v_0$からのグラフ上の距離が小さいものを優先的に選択する。

iii. 追加できるノードがなくなるまで、iiを繰り返す。

iv. V_{scaffold}を scaffold 配列として保存する。

v. i～ivをv_0となるノードの候補がなくなるまで繰り返す。

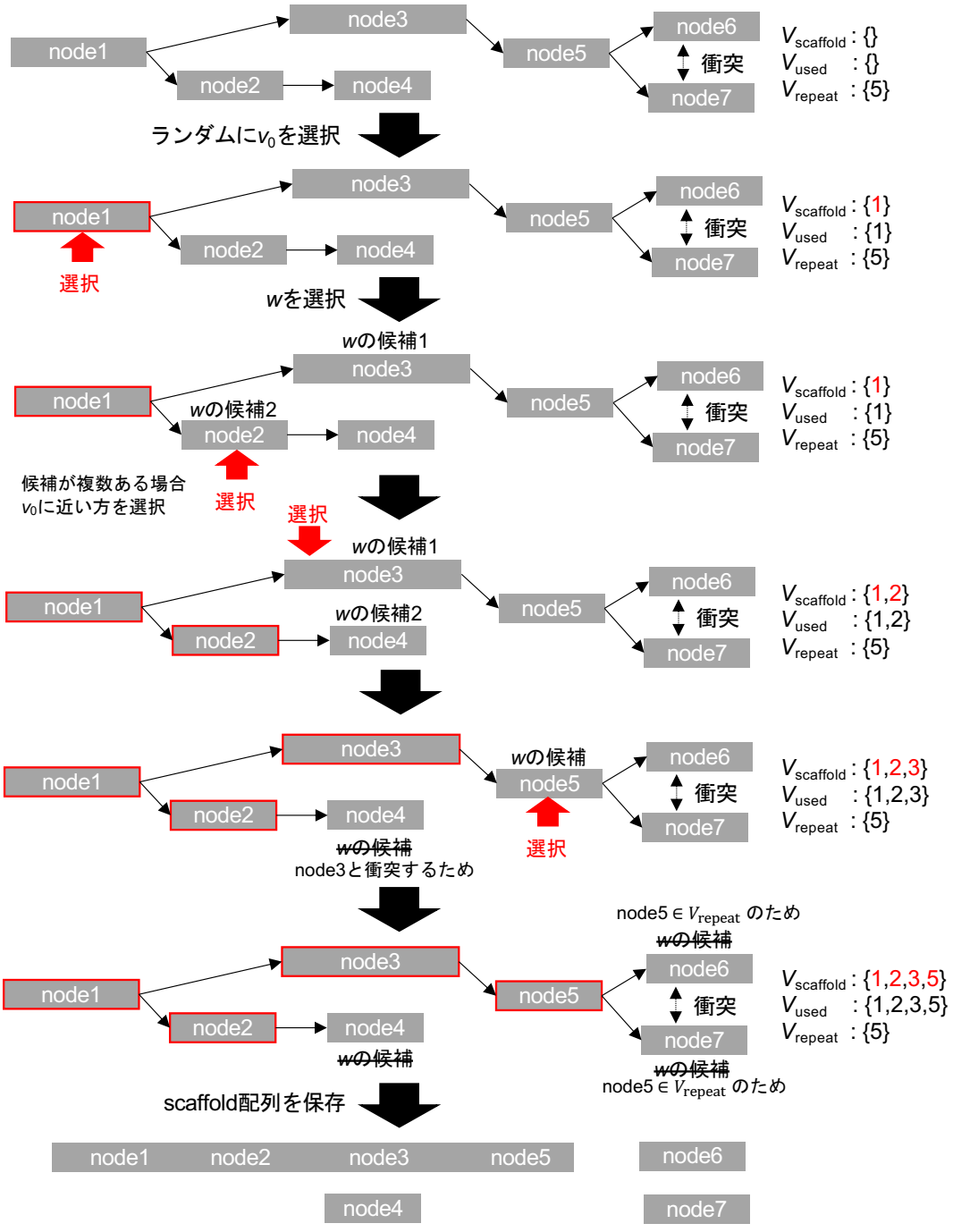

図 17　scaffold 構築

2.5 Consensus scaffolding by Hi-C

Consensus scaffolding の第二ステップでは、Hi-C による scaffolding を行い、染色体レベルで繋がった consensus scaffold を構築する。以下に手順を述べる(図 18)。

i. Hi-C のコンタクトマップ、ロングリード(および Pair-End リード)を使用して、入力の consensus scaffold 内のミスアセンブリを検出し、修正を行う(詳細は Consensus scaffold 内のミスアセンブリの修正の節を参照)。

ii. Consensus scaffold の 5'末端、3'末端をノード、scaffold 端間の Hi-C の link をエッジとしたグラフを作成する。エッジの重みは、入力の scaffold の端から距離 L の領域間を架橋する Hi-C リードペアの数として計算する。重みが閾値（default: 50 [L < 100 kb]、100 [L ≧ 100 kb]）以下のエッジは除去する。

iii. 重みが最大のエッジ e を検出する。

iv. エッジ e がエラーエッジでないかを確認する。Hi-C のコンタクトマップによる確認(詳細は Hi-C によるエラーエッジ除去の節を参照)とロングリードによる確認(詳細は Long read によるエッジの確認の節を参照)の両方を行い、その論理和をとることでエッジがエラーでないかを判定する。

v. ivでエッジ e がエラーと判定された場合、エッジ e を除去する。正しいと判定された場合、エッジ e の始点と終点のノードを接続する。

vi. iii〜vをグラフが変化しなくなるまで繰り返す。

vii. ii〜viを、L を徐々に増やしながら、繰り返す。10 kb〜100 kb までは 10 kb 刻み、100 kb〜1 Mb までは 100 kb 刻みで L を増加させる。L を徐々に増やしながら scaffold を繋ぎ、段階的に長い scaffold を構築していくことで、局所的な Hi-C の link と全体的な Hi-C の link を両方とも考慮できるようにしている。

viii. ivでのロングリードによるエッジの確認はなしで、ii〜viiをもう一度行う。これは scaffold 間にロングリードでは超えられないような長いリピート領域が存在しても、scaffold を繋げられるようにするために行う。

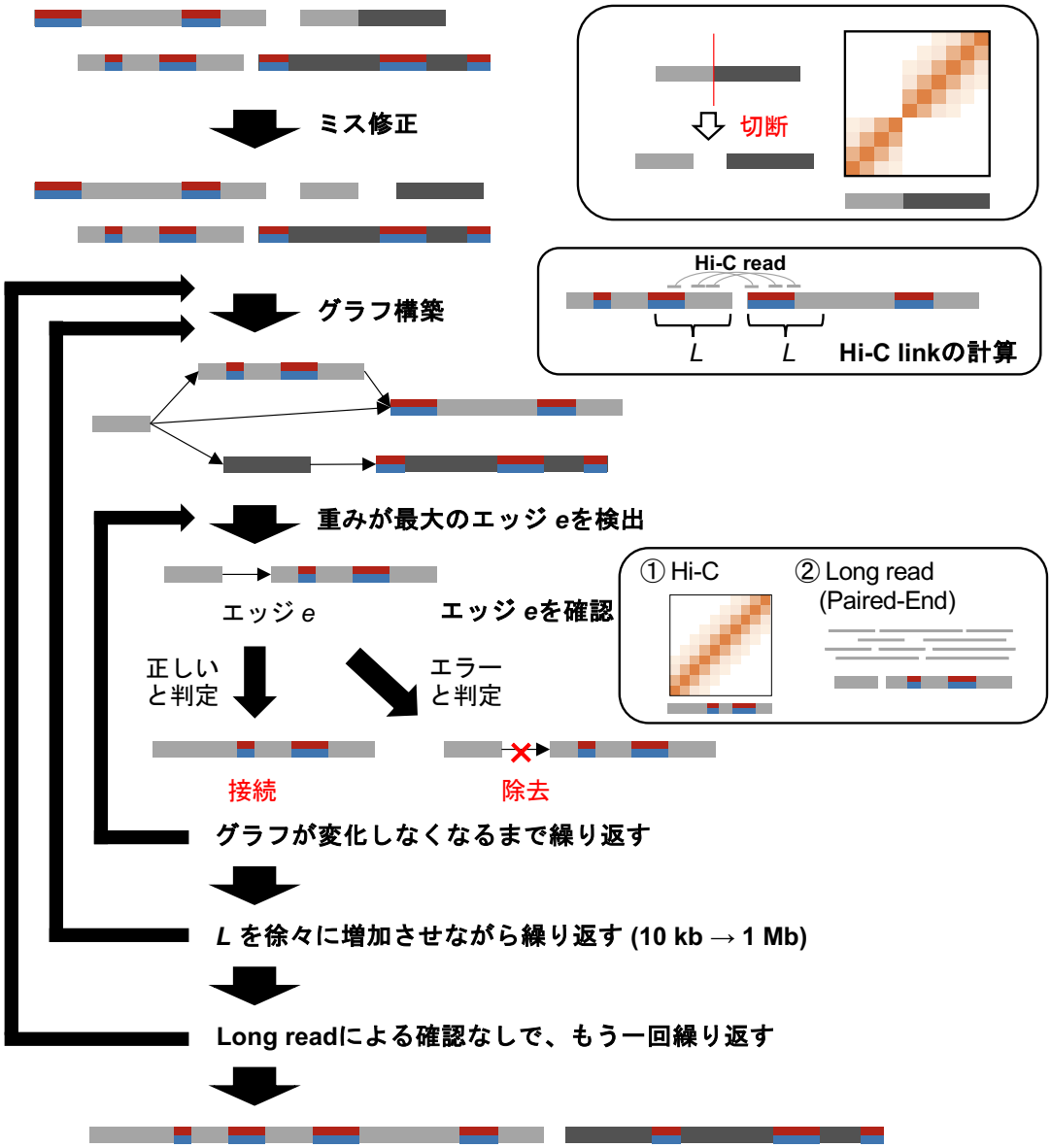

図 18　Consensus scaffolding by Hi-C

Consensus scaffold 内のミスアセンブリの修正

Hi-C のコンタクトマップ、ロングリードの coverage を用いて、入力の consensus scaffold 内のミスを検出し、修正を行う。Scaffold 長が短すぎると Hi-C コンタクトマップによりミスを検出するのは困難なため、300 kb 未満の scaffold のミス検出、修正は行わない。以下に手順を述べる。

i. 平均のコンタクトマップを作成

長さが 300 kb 以上の scaffold を用いて、平均のコンタクトマップを作成する。Scaffold を bin-size = 100 kb で区切り、bin 間を架橋する Hi-C リードペア数を計算し、bin 間の距離ごとに平均 Hi-C 架橋数を算出する。

ii. 実際のコンタクトマップを作成

対象の scaffold のコンタクトマップを bin-size = 100 kb で作成する。

iii. Mis-assembly score の計算 (図 19a)

平均と実際のコンタクトマップの差を mis-assembly score として計算する。ミスアセンブリの箇所では、その両側間を架橋する Hi-C リードペア数が少なくなるため、コンタクトマップでは対角線付近の架橋数が少なくなる傾向がある。そこで、対角線の右下の三角形の領域(10×10)で平均と実際のコンタクトマップの差を計算し、mis-assembly score とする。以下の式を用いて mis-assembly score は計算する。

$$\text{score} = \sum_{k=1}^{10} \frac{(\mu_{target,k} - \mu_{all,k})^2}{\mu_{all,k}}$$

$\mu_{target,k}$：対角線から距離 k のピクセルの実際のコンタクトマップの平均架橋数

$\mu_{all,k}$：対角線から距離 k のピクセルの平均のコンタクトマップの平均架橋数

ただし、平均架橋数を算出する際は、架橋数が閾値 *th* (default = 10000)より多い場合、架橋数を *th* として計算する。これはリピート領域などにより Hi-C の架橋数が異常に多い領域が原因で平均架橋数が多くなる影響を低減するためである。

iv. Mis-assembly score のピークを検出

iiiで計算した mis-assembly score のピークを検出する。

v. ピークの閾値 *P* の決定 (図 19b)

Mis-assembly score が大きい箇所はミスアセンブリの箇所と考えられる。しかし、mis-assembly score のピーク値がいくつ以上ならミスアセンブリと判定するか閾値を決定する必要がある。既存ツールの 3D-DNA では、この閾値に固定値を用いており、それが原因で Hi-C のデータ量が少ない場合にミスでない領域もミスと判定し切断してしまうという問題が発生している。そこで、本ツールではピークの閾値 *P* を自動で決定することでその問題に対処している。ivで検出したピークのうち、閾値 *P* 以上のピークの位置で scaffold を分け、各位置で分離度 *S* を計算する。閾値 *P* を 0 からピークの最大値まで変化させ、分離度 *S* の平均が最大になるように、閾値 *P* を決定する。

vi. 切断するかの判定

vで決定した閾値以上の各ピークについて、分離度 *S*、*R* を計算し、Hi-C によるエラーエッジ除去の節のiiiの条件を満たすとき、ミスアセンブリと判定する。

vii. breakpoint の決定、scaffold 切断

コンタクトマップは bin-size = 100 kb で作成しているため、切る場所(breakpoint)の正確な位置を決定することはできない。そこで、viで検出したミスアセンブリ箇所の 200 kb の領域でロングリードの coverage 計算し、coverage が最小となる位置を breakpoint として決定する。最後に、決定した breakpoint の位置で scaffold を切断する。

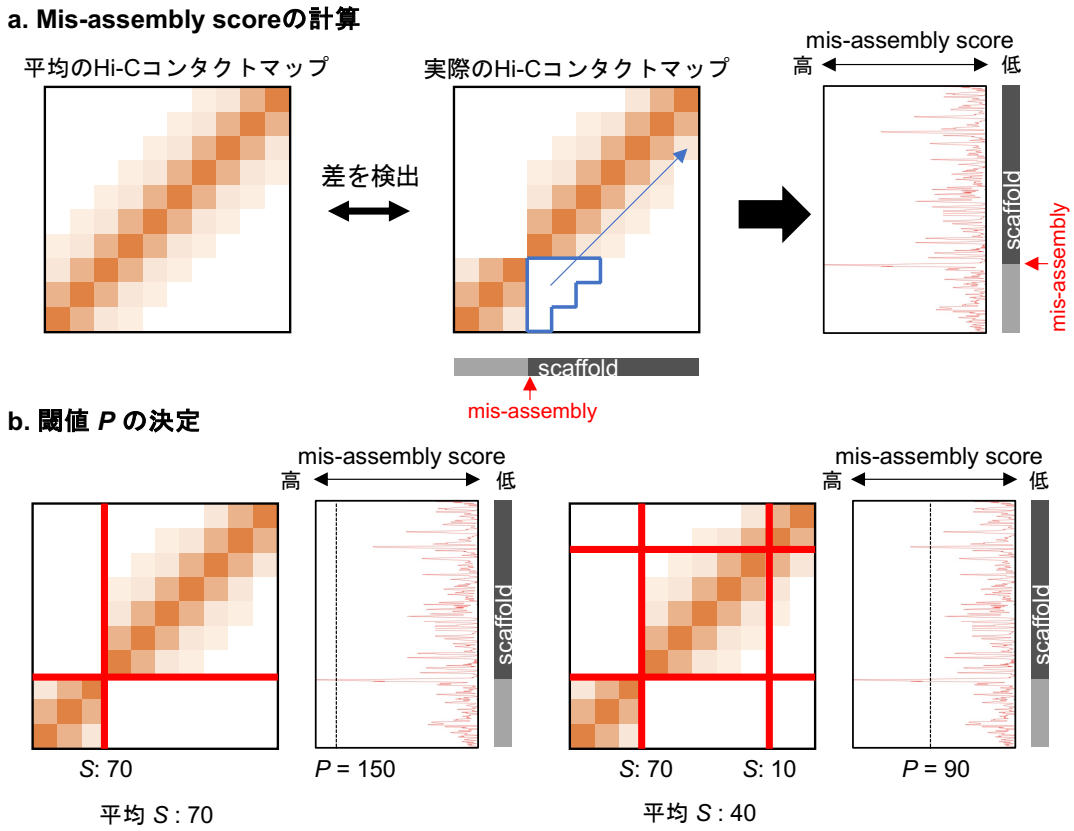

図 19　ミス修正

a. Mis-assembly score の計算の模式図。平均と実際のコンタクトマップの差を mis-assembly score として計算する。b. 閾値 *P* の決定の模式図。図の例では、閾値 *P* = 150 のとき、分離度 *S* の平均は 70、閾値 *P* = 90 のとき、分離度 *S* の平均は 40 となっている。分離度 *S* の平均が最大になるよう閾値 *P* を決定するため、この例では *P* = 150 が採用される。

Long read によるエッジの確認

二つのノード *u*、*w* を接続するエッジ *e* を、ロングリード(および Pair-End リード)を用いて以下の手順でエラーでないかを確認する(図 20)。

i. ロングリード(と Pair-End リード)の link を確認

ノード *u*、*w* 間のロングリード(および Pair-End リード)の link 数が閾値 *minLink* (default = 0)より多い場合、エッジ *e* は正しいと判定し、確認を終了する。

ii. 他の向きでロングリード(と Pair-End リード)の link を確認

他の向き、すなわちノード *u*、*w*'、ノード *u*'、*w*、ノード *u*'、*w*'間のロングリード(および Pair-End リード)の link 数を計算する。ただし、ノード *u*'、*w*'とはそれぞれノード *u*、*w* の scaffold の反対側の端を表すノードのことである。他の向きのノードペア間の link 数が *minLink* より多い場合、その二つのノードを繋ぐエッジを、エッジ *e* の代わりに採用し、Hi-C Scaffolding 時に用いる。条件を満たす向きが複数ある場合、link 数が最大の向きを採用する。

iii. Scaffold 端のミス検出

Scaffold の端 300 kb の領域で、ミスアセンブリがないかロングリード(および Pair-End リード)のマッピング情報をもとに確認し、ミスアセンブリがある場合ミスを修正する。Scaffold の端はミスアセンブリがあることが多く、それが原因で scaffold 間に link がない問題に対処するためにこの処理を行う。

はじめに、マッピング情報をもとにミス候補領域の特定を行う。Paired-End リードについては､各リードで start 位置(リードがマッピングされた位置)と end 位置(start 位置 + insert 長 tolerance)を計算する。ただし、insert 長 tolerance は

$$tolerance = a + 3 \times d$$

で計算する。ここで、aはライブラリの insert 長の平均、dは、ライブラリの insert 長の標準偏差である。ロングリードについては、start 位置は scaffold へのアライメント領域の端として計算する。End 位置は start 位置に scaffold へのアライメント領域と、他の scaffold へのアライメント領域との間の距離を足して計算する。ミスアセンブリは、start 位置と end 位置の間に位置すると考えられるため、start 位置と end 位置の間の領域をミス候補領域として検出する。

次に、ミスアセンブリ候補領域内で breakpoint を決定する。ミスアセンブリ位置では coverage が低下する傾向があるため、各ミスアセンブリ候補領域内で coverage が最小の位置を breakpoint として決定し、scaffold を切断する。

ミスアセンブリが確認された場合、ミスを修正しエッジ *e* を正しいと判定する。ミスアセンブリを確認できなかった場合、エッジ *e* はエラーエッジと判定する。

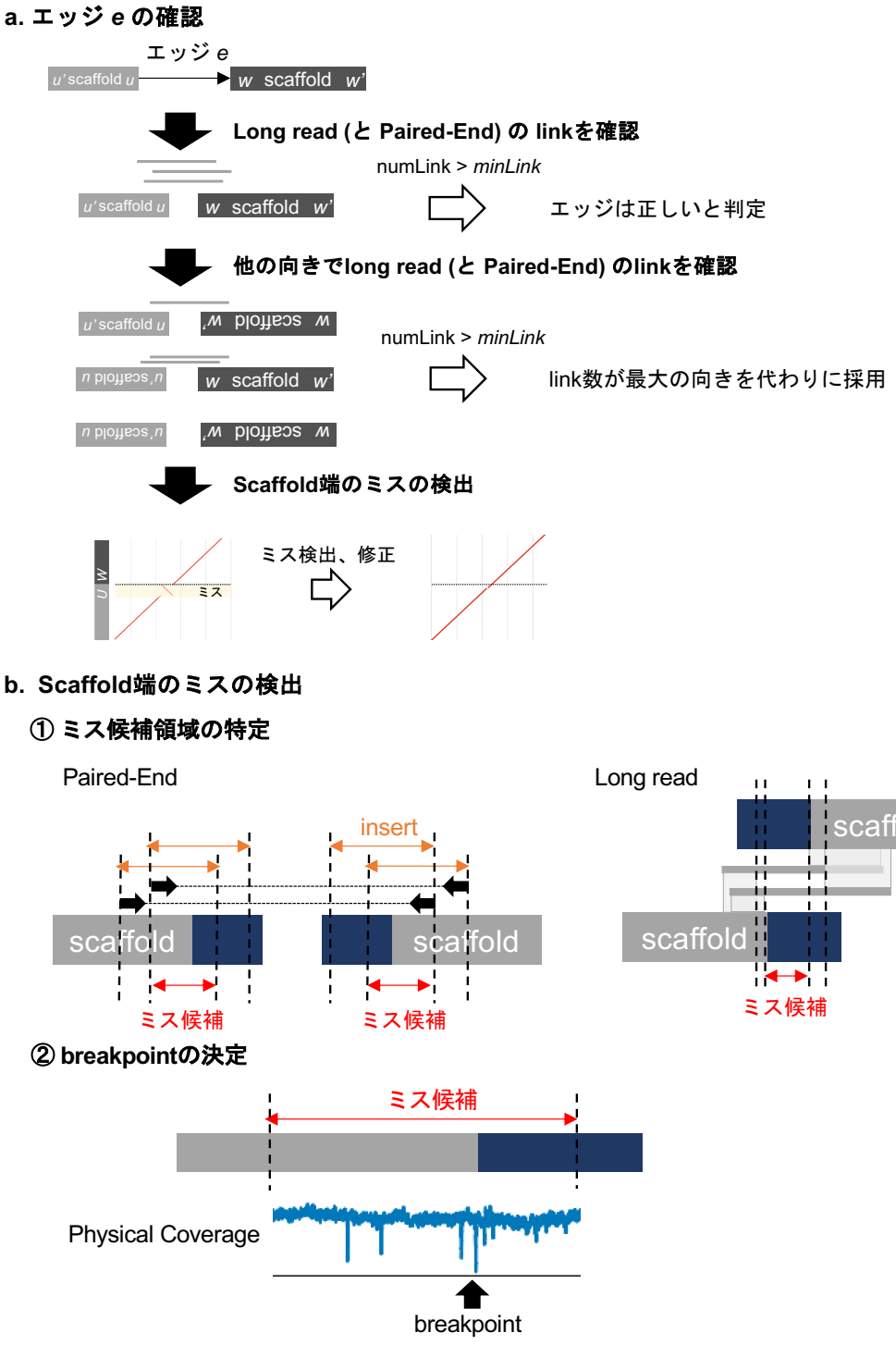

図 20　Long read によるエッジの確認

2.6 Phasing

Phasing では、Consensus scaffolding の結果をハプロタイプブロックに分割し、ロングリード(および Pair-End リード)、Hi-C を用いて phasing を行う。Consensus scaffolding 結果の各 scaffold に対して、以下の処理を行う（図 21)。

i. ヘテロブロックの特定

2.3 Merge haplotype で保存した情報を用いて、Consensus scaffolding の結果をハプロタイプブロックに分割する。各ハプロタイプブロックは、対応するハプロタイプの contig のペア(Haplotype0、Haplotype1)で構成されている。Haplotype0 = Haplotype1 のブロックをホモブロック、Haplotype0 ≠ Haplotype1 のブロックをヘテロブロックとして特定する。

ii. 全てのヘテロブロック間の link 数を集計

iで特定した全ヘテロブロック間のロングリード(および Pair-End リード)と Hi-C の link 数の合計を Parallel-Path、Cross-Path で別々に集計する。ここで、i 番目のヘテロブロックの Haplotype0 を$\mathrm{Haplotype}_i0$、Haplotype1 を$\mathrm{Haplotype}_i1$とすると、i 番目のヘテロブロックと j 番目のヘテロブロック間の Parallel-Path を支持する link 数は、$\mathrm{Haplotype}_i0$と$\mathrm{Haplotype}_j0$、$\mathrm{Haplotype}_i1$と$\mathrm{Haplotype}_j1$間の link 数の合計、Cross-Path を支持する link 数は、$\mathrm{Haplotype}_i0$と$\mathrm{Haplotype}_j1$、$\mathrm{Haplotype}_i1$と$\mathrm{Haplotype}_j0$間の link 数の合計で計算する。

iii. Parallel-Path と Cross-Path の link 数の差が大きいペアから順に解決

Parallel-Path と Cross-Path を支持する link 数の差が最大のヘテロブロックのペアを結合する。Cross-Path の方が支持する link 数が多い場合、片方のヘテロブロックの Haplotype0 と Haplotype1 を入れ替えた後結合する。

iv. iiiを Parallel-Path と Cross-Path を支持する link 数の差が 1 以上のペアがなくなるまで繰り返す。

v. Phasing の後処理

iiiで結合できなかったヘテロブロックとヘテロブロック間で scaffold を切断する。ただし、ヘテロブロック内に未結合のヘテロブロックがある場合、その未結合のヘテロブロックの位置での切断は行わない。

vi. Phasing 結果をもとに、2 つのハプロタイプ配列を構築し、結果を出力する。

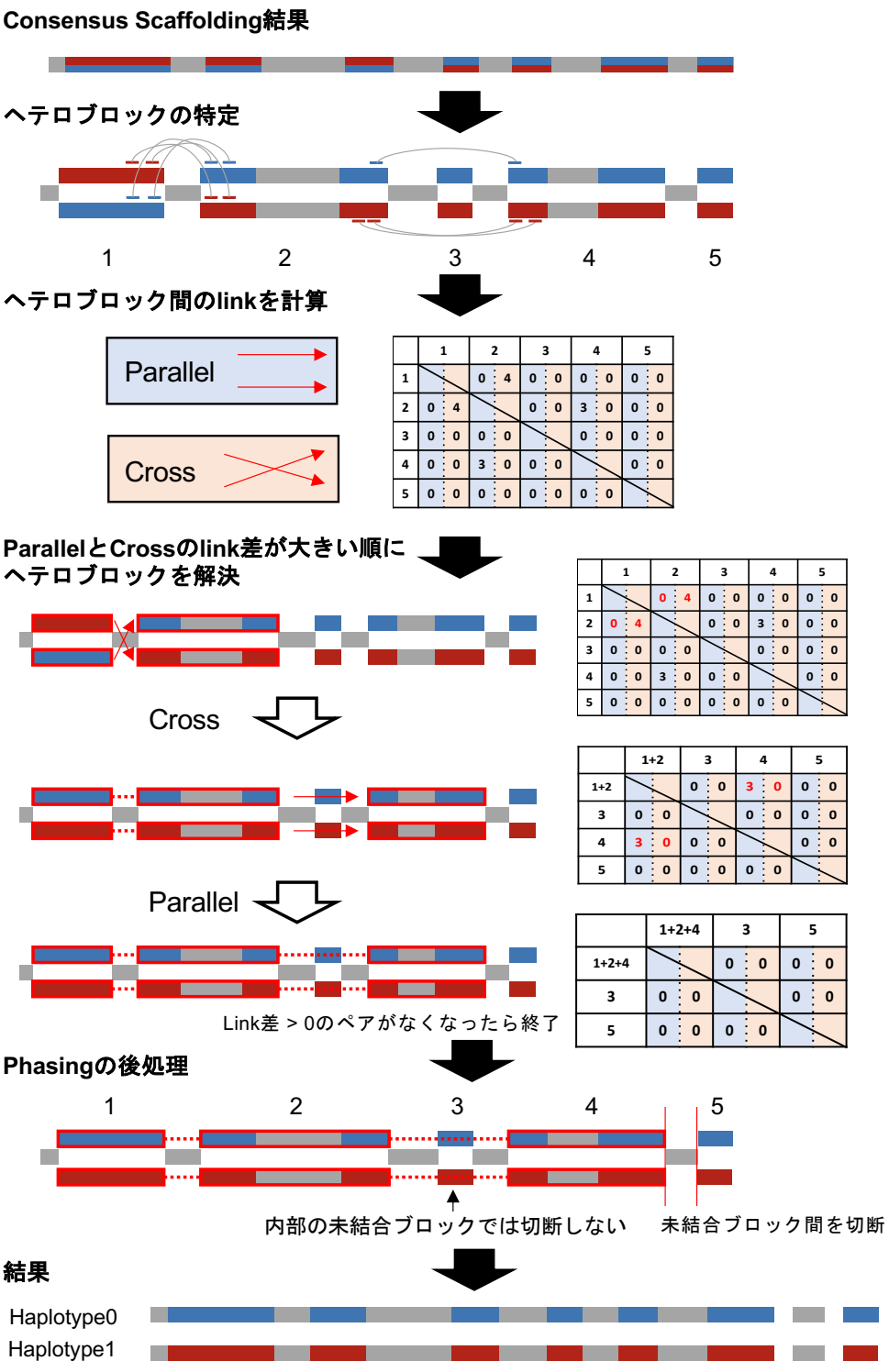
Consensus Scaffolding結果
ヘテロブロックの特定
1
2
3
4
5
ヘテロブロック間のlinkを計算
Parallel
Cross
ParallelとCrossのlink差が大きい順に
ヘテロブロックを解決
Cross
Parallel
Link差 > 0のペアがなくなったら終了
Phasingの後処理
1
2
3
4
5
内部の未結合ブロックでは切断しない
未結合ブロック間を切断
結果
Haplotype0
Haplotype1

図 21 Phasing

第 3 章　ベンチマーク

本章では、前章で説明したアルゴリズムに基づき開発された GreenHill の性能を評価するため、様々な生物種のシミュレーションおよび実データで行ったベンチマークの結果を示す。

3.1 ベンチマーク方法

3.1.1 使用データ

ベンチマークに用いた生物種は以下の通りである。

・*Caenorhabditis elegans*

モデル生物として広く用いられている線虫のデータ。２つの系統の株を用いて擬似的な二倍体のシミュレーションデータを作成し、ベンチマークに用いた。

・*Drosophila melanogaster*

モデル生物として広く用いられているキイロショウジョウバエのデータ。Hi-C シミュレーションデータ、ロングリード実データを用いてゲノムを構築し、完成度の高いリファレンス配列で結果を評価した。

・*Bos indicus*×*Bos taurus* (Cow)

Bos indicus(コブウシ)と *Bos taurus*(ウシ)の交雑種のデータ。先行研究[51]で使用されている実データを使用しベンチマークを行った。評価には両親のショートリードデータを用いた。

・*Taeniopygia guttata* (Zebra finch)

鳥類の一種であるキンカチョウのデータ。脊椎動物ゲノムプロジェクト(Vertebrate Genomes Project、VGP[63])でゲノムが決定され、公開されている実データを使用した。評価には両親のショートリードデータを用いた。

・*Melopsittacus undulatus* (budgerigar)

鳥類の一種であるセキセイインコのデータ。VGP でゲノムが決定され、公開されている実データを使用した。評価には両親のショートリードデータを用いた。

・*Diceros bicornis* (black rhinoceros)

哺乳類の一種であるクロサイのデータ。VGP でゲノムが決定され、公開されている実データを使用した。評価には両親のショートリードデータを用いた。

・*Acipenser ruthenus* (sterlet)

魚類の一種であるコチョウザメのデータ。VGP でゲノムが決定され、公開されている実データを使用した。評価には両親のショートリードデータを用いた。

使用したシーケンスデータの詳細を表 2 に示す。

表 2　シーケンス情報

生物種	ライブラリ	平均リード長 (bp)	合計サイズ (bp)	推定 coverage (×)
線虫	Paired-End	250	7,943,225,500	80
	PacBio CLR	10,482	8,028,062,739	80
	Hi-C	150	5,999,999,890	60
ショウジョウバエ	PacBio CLR	17,140	30,001,763,138	200
	PacBio HiFi	24,429	5,995,780,057	40
	Hi-C	150	8,399,986,576	56
ウシ	Paired-End	150	234,044,926,410	86
	PacBio CLR	3,562	275,464,947,718	100
	Hi-C	80	32,534,720,960	12
キンカチョウ	10X	102	107,638,065,920	105
	PacBio CLR	9,025	102,219,439,023	100
	PacBio HiFi	11,421	39,367,893,111	39
	Hi-C	150	95,764,922,700	94
セキセイインコ	10X	150	97,167,010,234	87
	PacBio CLR	18,291	75,354,296,130	67
	Hi-C	150	104,410,326,300	93
クロサイ	PacBio HiFi	14,737	96,526,923,246	32
	Hi-C	150	226,606,277,700	76
コチョウザメ	PacBio HiFi	13,546	111,606,998,728	63
	Hi-C	150	259,186,416,132	147

推定 coverage は、*k*-mer 出現頻度分布による解析から推定したゲノムサイズを用いて算出した。

各生物種のゲノムの特徴を把握するために、*k*-mer 出現頻度分布による解析を行った。*k*-mer 出現頻度分布とは、*k* 文字の部分文字列(*k*-mer)を各リード配列から 1 bp ずつずらしながら取得し、全リード配列中の *k*-mer の種類数と出現回数を集計し、ヒストグラムにしたものである。*k* が十分大きければ、ほとんどの *k*-mer は、ゲノム中の 1 カ所に由来し、その出現回数はリードの coverage と一致するため、*k*-mer 出現分布では平均 coverage にピークを持つ分布が観察される。二倍体生物でヘテロ接合度の高いゲノムの場合、相同染色体間で変異のあるヘテロな領域から 2 種類の *k*-mer が得られ、それらの出現回数はリードの coverage の半分程度になるため、*k*-mer 出現分布は平均 coverage の半分の値にもピークを持つ 2 峰性の形をとる。これらの特徴により、ゲノムサイズは、リード中のエラーに由来する出現頻度が低い *k*-mer を除いた全 *k*-mer 数 / ホモなピーク値を求めることで、推定することができる。また、ヘテロ接合度は、ヘテロな領域のピーク値とホモな領域のピーク値を比較することで推定可能である。

リード中の *k*-mer の出現回数のカウントは、jellyfish[64]を用いて行い(*k*＝21)、ゲノムサイズ、ヘテロ接合度の推定は Genomescope[65]を用いて行った。結果を表 3、図 22 に示す。入力のリードには、線虫、ウシは、Paired-End リード、ショウジョウバエ、キンカチョウ、クロサイ、コチョウザメは、PacBio HiFi リード、セキセイインコは 10X リードを用いた。ウシのゲノムサイズとヘテロ接合度の推定値については、Genomescope によるモデルのフィッテイングがデータと大きく異なり信頼性が低かったため、先行研究[56]による推定値を代わりに用いた。

表 3　推定ゲノムサイズとヘテロ接合度

生物種	入力データ	推定 ゲノムサイズ	推定 ヘテロ接合度 (%)
線虫	Paired-End	99.3 Mb	0.31
ショウジョウバエ	PacBio HiFi	139 Mb	0.74
ウシ	Paired-End	2.74 Gb	0.69-0.93
キンカチョウ	PacBio HiFi	1.01 Gb	1.47
セキセイインコ	10X	1.12 Gb	1.04
クロサイ	PacBio HiFi	2.98 Gb	0.21
コチョウザメ	PacBio HiFi	1.76 Gb	0.58

Genomescope により、各生物種の推定ゲノムサイズ、ヘテロ接合度を計算。ただし、ウシは先行研究による推定値を代わりに用いた。

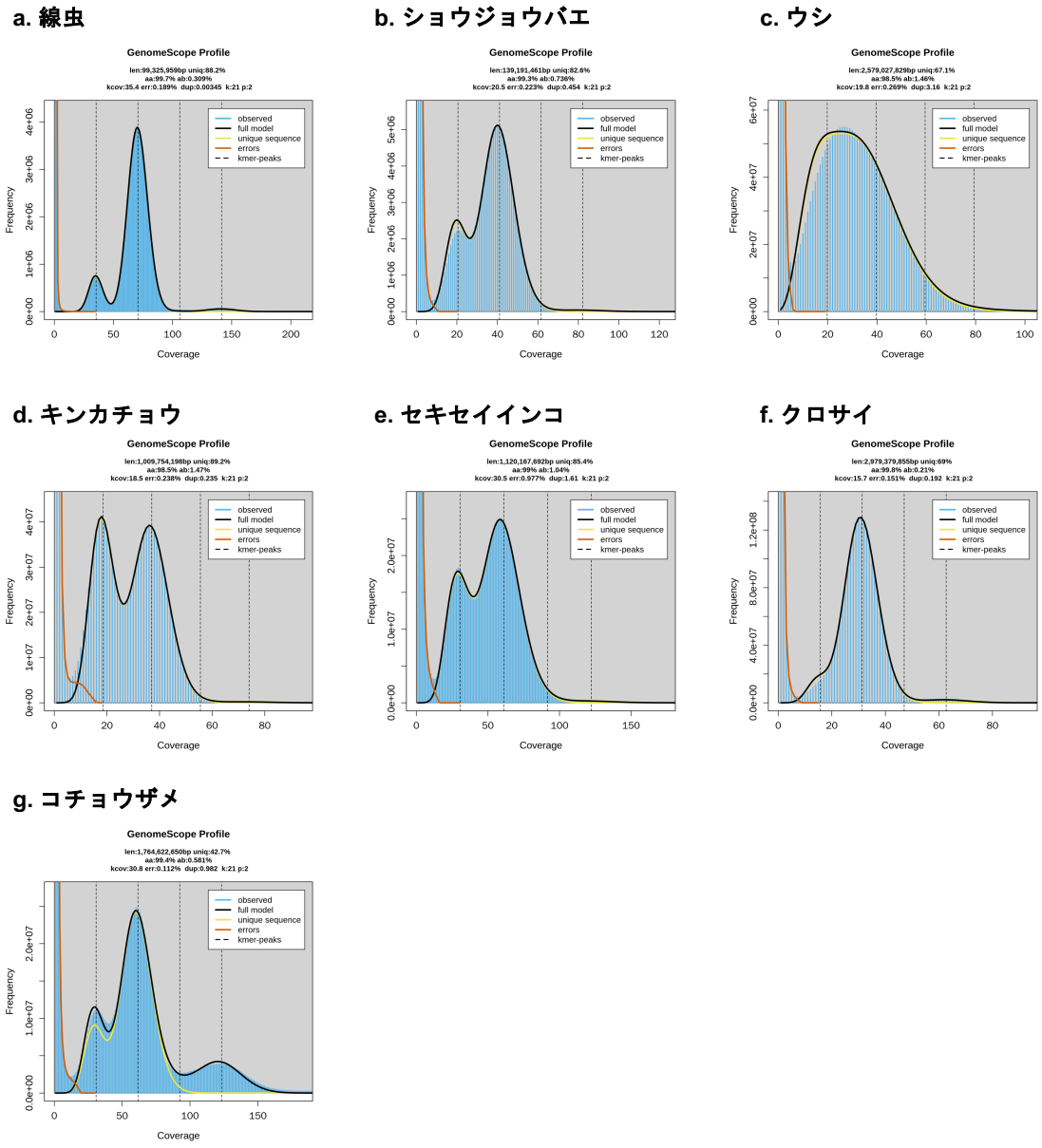

図 22　Genomescope による解析結果

各生物種の Genomescope による解析結果。横軸が *k*-mer の出現回数(coverage)、縦軸が *k*-mer の種類数を表す。青色の棒グラフが実際に観察された *k*-mer のヒストグラム、黒線が Genomescope によりフィッティングされたモデル、黄線がユニークと推定されたもの、赤線がリードのエラー由来と推定されたもの、黒い縦点線は、推定された *k*-mer のピークを示している。

3.1.2 入力 contig の作成

様々な入力に対する GreenHill の性能を評価するため、Paired-haplotype style の入力として Platanus-allee によるアセンブル結果、Pseudo-haplotype style の入力として FALCON-Unzip によるアセンブル結果、Haplotype-ignorant style の入力として Canu によるアセンブル結果を入力 contig として用いた。また、HiFi リードが入力の場合、HiCanu、hifiasm によるアセンブル結果を入力 contig として用いた。各ツールの実行方法を以下に示す。

・Platanus-allee

Platanus-allee[55] v2.2.2 の scaffolding 機能を Hi-C データを扱えるように改良し、v2.2.2-modifed version として実装したもの(https://github.com/ShunOuchi/GreenHill でコードは公開している)を使用した。v2.2.2 との主な変更点は、ヘテロ領域由来のクロス構造を解く処理に Hi-C データも使用するようにした点である。Hi-C データも使用することでより連続性の高い contig を構築できるようになっている。

入力ファイル、マルチスレッドのパラメータ以外はデフォルトのパラメータで実行した。Platanus-alee では、”assemble”コマンドで contig assembly、”divide”コマンドで contig 内のミスの検出・切断、”phase”コマンドでハプロタイプを考慮した scaffolding を行うが、”assemble”、”divide”コマンドの入力には、Paired-End リード、”phase”コマンドの入力には、Paired-End リード、PacBio ロングリード、Hi-C リードを用いた。ただし、キンカチョウのベンチマークでは、Paired-End リードが入手できなかったため、10X リードを Paired-End リードの代わりに使用した。

・FALCON-Unzip

FALCON-Unzip[32] (v1.2.0)を以下のパラメータで実行した。

“length_cutoff = -1; length_cutoff_pr = 5000; pa_daligner_option = -e0.76 -l1200 -k18 -h70 -w8 -s100; ovlp_daligner_option = -k24 -h1024 -e.95 -l1800 -s100; pa_HPCdaligner_option = -v -B128 -M24; ovlp_HPCdaligner_option = -v -B128 -M24; pa_HPCTANmask_option = -k18 -h480 -w8 -e.8 -s100; pa_HPCREPmask_option = -k18 -h480 -w8 -e.8 -s100; pa_DBsplit_option = -x500 -s400; ovlp_DBsplit_option = -s400; falcon_sense_option = --output-multi --min-idt 0.70 --min-cov 3 --max-n-read 400 --n-core 24; overlap_filtering_setting = --max-diff 100 --max-cov 150 --min-cov 3 --n-core 24”

ウシとキンカチョウについては FALCON-Unzip によるアセンブル結果を NCBI データベースからダウンロードして使用した（ウシ：GCA_012069665.1、GCA_012070425.1、キンカチョウ：GCA_012069585.1、GCA_012069535.1)。ただし、キンカチョウの結果は、primary contig 中のハプロタイプ重複を除去するために、FALCON-Unzip 結果に purge haplotig[66]を実行した結果である。

・Canu

Canu[30] (v2.1.1)を以下のオプションを使用して実行した。

“corOutCoverage = 200 batOptions = -dg 3 -db 3 -dr 1 -ca 500 -cp 50"

corOutCoverage オプションは、エラー修正したリードの coverage の上限値を指定するオプションで、デフォルトでは実行時間を短縮するため 40 に設定されているが、より正確にゲノムを構築するために 200 に設定した。また、” batOptions = -dg 3 -db 3 -dr 1 -ca 500 -cp 50”は、リード間のオーバラップ検出の基準をより厳密にするために設定した。このオプションは、高ヘテロ接合度のゲノムで使用が推奨されているオプションで、ヘテロな領域を潰しハプロタイプ間の違いを無視した配列ができるのを避け、２つのハプロタイプをそれぞれ別々に構築できるようにするため使用した。

・HiCanu

HiCanu[35](v2.1.1)を、入力ファイル、マルチスレッドのパラメータ以外はデフォルトのパラメータを使用して実行した。

・Hifiasm

Hifiasm [34](v0.16.1)を、入力ファイル、マルチスレッドのパラメータ以外はデフォルトのパラメータを使用して実行した。

作成した入力 contig の統計値を表 4 に示す。

表 4　入力 contig の統計値

生物種	使用ツール	全長 (bp)	最大長 (bp)	N50 (bp)	QV	スイッチエラー率
線虫	Platanus-allee	201,915,293	2,357,661	480,746	37.26	1.08
	FALCON-Unzip	189,700,890	4,713,319	441,947	31.33	2.16
	Canu	133,218,907	2,976,042	608,865	30.12	4.14
ショウジョウバエ	Canu	296,228,027	12,908,400	680,260	35.12	4.16
	HiCanu	327,107,205	25,974,267	9,646,753	67.95	0.04
	Hifiasm (p_utg)	329,314,953	12,900,022	1,808,138	64.55	0.03
ウシ	Platanus-allee	5,716,453,794	4,343,604	446,852	31.82	6.01
	FALCON-Unzip	5,121,908,670	99,126,329	4,551,535	41.84	0.18
キンカチョウ	Platanus-allee	2,169,120,089	5,688,294	470,572	35.94	0.50
	FALCON-Unzip	1,976,041,882	20,242,265	931,146	35.59	0.81
	Canu	1,928,989,367	5,546,997	392,153	35.34	2.09
	Hifiasm (p_utg)	2,184,967,032	23,325,922	3,035,032	50.33	0.01
セキセイインコ	FALCON-Unzip	2,056,688,110	49,947,430	4,453,551	40.22	0.30
クロサイ	Hifiasm (p_utg)	6,314,336,883	9,538,517	582,028	66.65	0.02
コチョウザメ	Hifiasm (p_utg)	3,861,485,552	10,699,564	1,682,164	60.74	0.02

全長、N50 は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された contig の合計長を表す。QV、スイッチエラー率は Merqury で計算した。

3.1.3 既存ツールの実行方法

GreenHill の性能を、FALCON-Phase[56]、hifiasm Hi-C mode[58]と比較した。

FALCON-Phase には Hi-C による scaffolding 機能はないため、他の Hi-C Scaffolding ツールと組み合わせた結果を比較として用いた。FALCON-Phase 論文[56]で使用されている方法を参考に、以下の手順で実行した(図 23)。

i. パージ

FALCON-Phase は、FALCON-Unzip 結果の形式しか入力として使用できないため、入力 contig が FALCON-Unzip によるアセンブル結果以外の場合、以下の方法で形式を変換した。はじめに、Purge_dups[67](v1.2.5)を用いて、入力 contig をパージし、purged primary contig と purged haplotig の二つの配列に分けた。Purge_dups で使用する coverage の閾値は、coverage のヒストグラムから手動で設定した。次に、各 haplotig を minimap2 で primary contig にマッピングし、haplotig の対応する領域の primary contig を特定した。最後に、primary contig、haplotig のヘッダーの名前を、FALCON-Unzip 形式(primary contig と haplotig の位置関係を示す名前)に自作のスクリプトを用いて変更した。

ii. FALCON-Phase round 1

入力 contig を FALCON-Phase(v1.2.0)を用いて、以下のパラメータで phasing した。

“min_aln_len = 3000, iterations = 1000000, output_format = pseudohap”

制限酵素認識サイトのパラメータは、入力の Hi-C ライブラリに対応する認識サイトを設定した（線虫、ショウジョウバエ : ”AAGCTT”、ウシ、コチョウザメ : ”GATC ”、キンカチョウ、セキセイインコ : ” GATC, GAATC, GATTC, GAGTC, GACTC”、クロサイ : ” GATC, GAATC, GATTC, GAGTC, GACTC, CTAAG, CTTAG, CTGAG, CTCAG, TTAA”）。

ウシとキンカチョウの FALCON-Unzip + FALCON-Phase の結果については、NCBI データベースからダウンロードして使用した（ウシ : GCA_012070465.1 、GCA_012070445.1、キンカチョウ : GCA_012069615.1、GCA_012069575.1）。

FALCON-Phase の結果として 2 つのハプロタイプ”phased.0”と”phased.1”が作成された。このステップは、FALCON-Phase を contig に対して実行し、phasing された contig を作成することで、次の Hi-C Scaffolding の精度を高めるために行った。

iii. Hi-C Scaffolding

FALCON-Phase の結果のうち”phased.0”を、Hi-C Scaffolding ツールを用いて scaffolding した。Hi-C Scaffolding ツールとしては、3D-DNA[22]と SALSA2[46]を用い

た。

・3D-DNA による Hi-C Scaffolding

はじめに Juicer[68] (v1.5.6)を用いて contig に Hi-C リードをマッピングした。パラメータは全てデフォルトのパラメータを使用し、内部のマッピングツールは Burrows–Wheeler Alignment (BWA) (v0.7.17-r1188)[69]を使用して実行した。Juicer が必要とする制限酵素サイトファイルは、”generate_site_positions.py”スクリプトを用いて作成した。複数の制限酵素を用いて作成された Hi-C ライブラリが入力の場合は、”digest_genome2.py”スクリプトを用いて BED ファイルを作成し[70]、”hic-pro2juicer.py”スクリプトを用いて BED ファイルを Juicer 用の制限酵素サイトファイルに変換した[70]。最後に、Juicer による Hi-C マッピング結果を元に 3D-DNA(v180922)で Hi-C Scaffolding を行った。3D-DNA は全てデフォルトのパラメータで実行した。

・SALSA2 による Hi-C Scaffolding

はじめに、Arima mapping pipeline (v100617)[71]に従い、contig に Hi-C リードをマッピングした。BWA で Hi-C リードをマッピングし、SAMtools(v1.3.1)[72]でマッピング結果を処理し、Picard(v1.141)[73]で PCR 重複の除去を行った。BAM 形式のマッピング結果は、BEDTools(v2.27.1)[74]の bamToBed コマンドで、SALSA2 の入力に必要な BED ファイルへ変換した。最後に、Arima mapping pipeline による Hi-C マッピング結果を元に SALSA2 で Hi-C Scaffolding を行った。SALSA2 は全てデフォルトのパラメータで実行した。

iv. FALCON-Phase round 2

Round 1 の FALCON-Phase の結果の”phased.1”と Hi-C Scaffolding 結果のペアを入力として再度 FALCON-Phase を実行した。FALCON-Phase は、FALCON-Unzip 形式しか入力として使用できないため、”phased.1”と Hi-C Scaffolding 結果の形式を以下の方法で変換した。はじめに、各”phased.1”の contig を minimap2 で Hi-C Scaffolding 結果にマッピングし、”phased.1” contig の対応する領域の Hi-C Scaffolding 結果を特定した。その後、“phased.1”と Hi-C Scaffolding 結果のヘッダーの名前を、FALCON-Unzip 形式(Hi-C Scaffolding 結果と”phased.1”の位置関係を示す名前)に自作のスクリプトを用いて変更した。

FALCON-Phase の実行時のパラメータは、FALCON-Phase round 1 と同じパラメータを用いた。

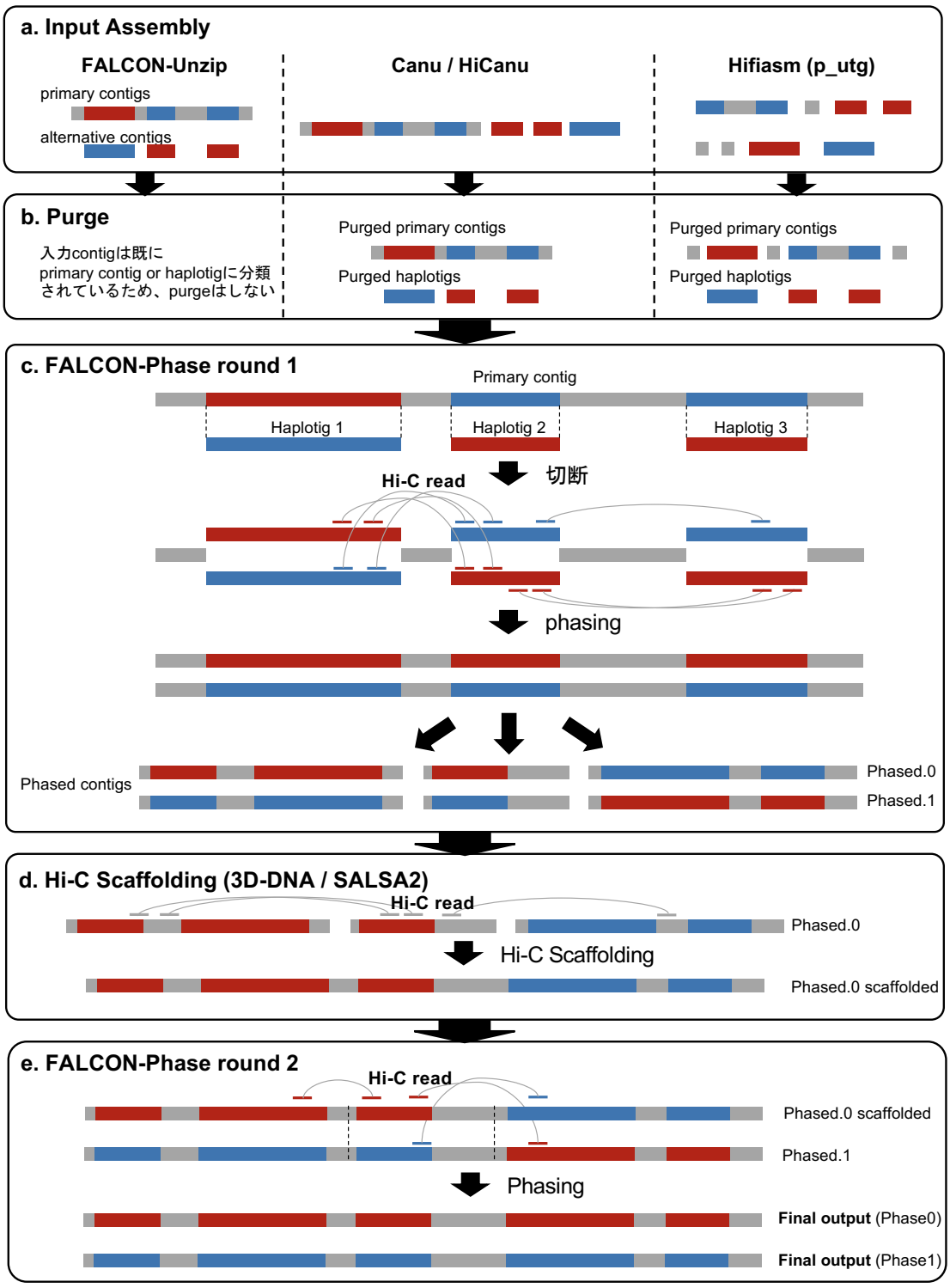

図 23　FALCON-Phase と Hi-C Scaffolding ツールの実行方法

HiFi データを使用したベンチマークでは、hifiasm Hi-C mode[58]も比較対象として用いた。Hifiasm Hi-C mode は全てデフォルトのパラメータで実行した。

3.1.4 評価方法

ゲノムアセンブル結果は、連続性、精度の２点で評価した。

連続性の評価

ゲノムアセンブル結果の連続性を、N50 を用いて評価した。N50 とは、scaffold 長の加重平均のことで、scaffold を長い順に並べ足していった際に scaffold 合計長の半分に達する scaffold 長のことである。N50 が大きいほど、長く連続した scaffold を構築できていることを表す。

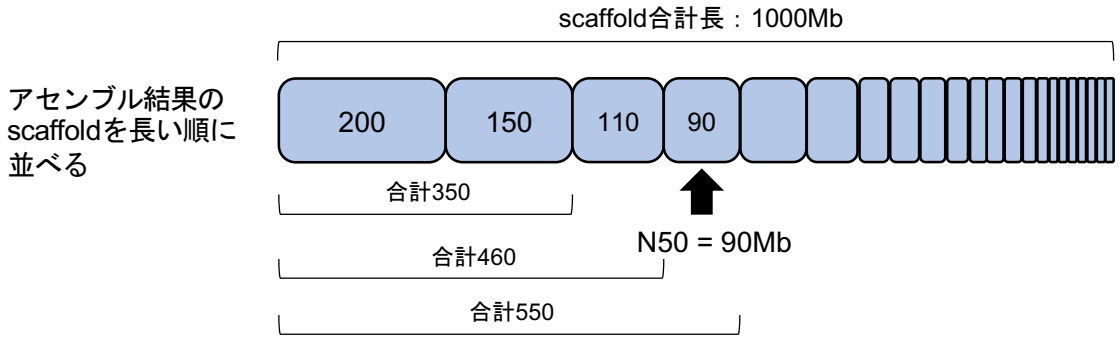

図 24　N50 の説明

精度の評価

ゲノムアセンブル結果の精度は、リファレンス配列にアライメントする方法または *k*-mer ベースによる方法を用いて評価した。

・リファレンス配列による評価

信頼性の高いリファレンス配列がある場合、リファレンス配列にアセンブル結果をアライメントすることで精度を評価した。以下に詳細な手順を述べる(図 25)。

i. Scaffold を固定長 (10 kb)の断片に区切る。

ii. 断片をリファレンス配列にアライメントする。

アライメントは、minimap2 を用いて計算した。アライメント結果のうち、ベストヒット、Identity > 95%、cover 率 > 90%の結果のみを次の処理に用いた。

iii. ミスアセンブリ数、スイッチエラー数を計算する。

ミスアセンブリ数は scaffold 内で距離が近い断片がリファレンス配列で離れた位

置にアライメントされる箇所をカウントして計算した。以下の二つの条件のうち、いずれかを満たすとき、ミスアセンブリと判定した。

・同じ scaffold 内の２つの断片が、リファレンス配列で異なる染色体にアライメントされる

・$|d1 - d2| > \frac{\min(d1,d2)}{2}$

ただし、*d1*：scaffold 内の２つの断片の距離
d2：リファレンス配列内の２つの断片の距離

スイッチエラー数は、scaffold 内で隣なり合う断片が異なるリファレンス配列にアライメントされる箇所をカウントした。ただし、ベストヒットが複数存在する場合はスイッチエラー検出には使用しなかった。

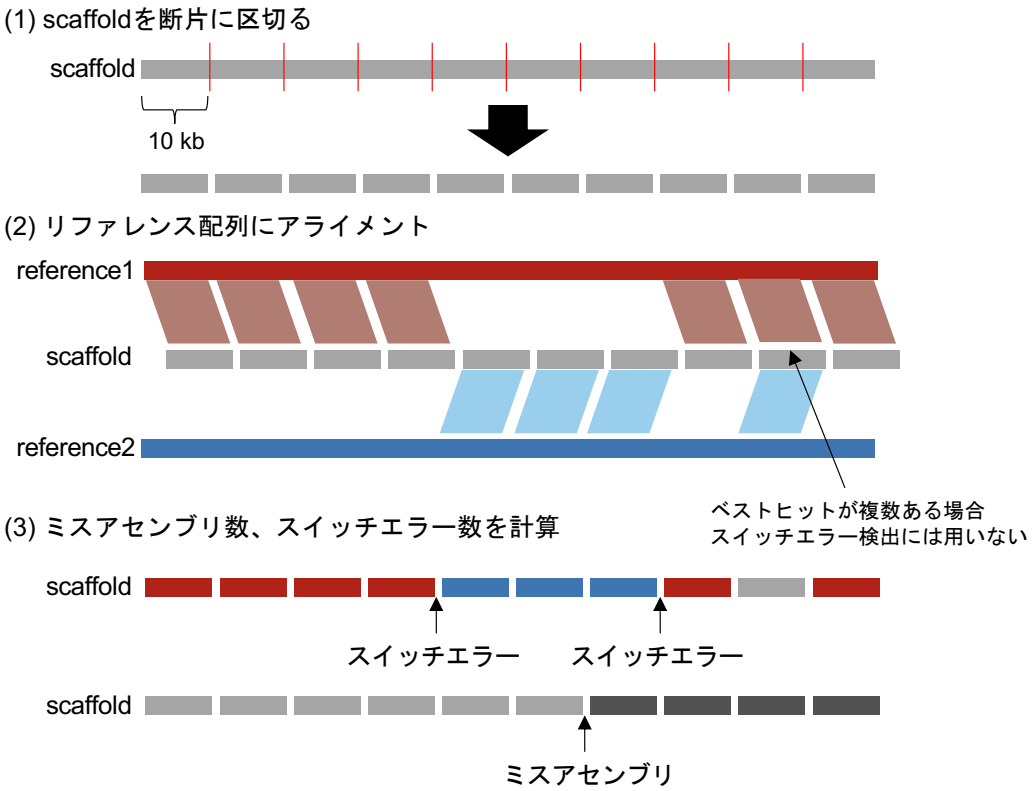

図 25　リファレンス配列による精度評価

・*k*-mer ベースによる評価

リファレンス配列がない場合、*k*-mer ベースの方法により、アセンブル結果を評価した(図 26)。以下の３つの指標を評価に用いた。

・Quality Value(QV)

アセンブル結果の塩基レベルの正確度(Quality Value、QV)を Merqury[75]を用いて計算した。Merqury ではまずアセンブル結果のみに存在しリードセットには存在しない *k*-mer をエラーと見なして、エラー率 E を以下の式で計算する。

$$E = 1 - (1 - \frac{K_{asm}}{K_{total}})^{\frac{1}{k}}$$

ただし、K_{asm}：アセンブル結果のみに存在する *k*-mer の数

K_{total}：アセンブル結果に存在する全 *k*-mer 数

そして、エラー率 E を元に広く使用されている Phred[76] quality score を QV として計算する。

$$QV = -10 \log_{10} E$$

QV が大きいほど、塩基レベルの精度が高いことを示す。

・スイッチエラー率

Merqury でスイッチエラー率を計算した。Merqury ではハプロタイプ固有の *k*-mer (hap-mer)を用いてスイッチエラー率を計算している。まず、両親のショートリードから *k*-mer をカウントし、母方固有の k-mer (hap-mer)、父方固有の *k*-mer (hap-mer)を計算する。そして、同じ scaffold 内で隣り合う hap-mer が異なる親由来である割合をスイッチエラー率として計算する。

・Phasing 精度

Merqury で特定した hap-mer を用いて、以下の式で Phasing 精度を計算した。

$$\text{Phasing 精度} = \frac{H_{major}}{H_{total}}$$

ただし、H_{major}：scaffold で多数派の親由来の hap-mer の数の合計

H_{total}：scaffold に存在する全 hap-mer 数の合計

これは scaffold 内の多数派の親由来の hap-mer の割合を示している。この値が高いほど、母方由来、父方由来のハプロタイプを分けて構築できており、phasing 精度が高いことを示す。注意すべきは、Phasing 精度とスイッチエラー率に相関はない点で、scaffold の中央に一つスイッチエラーがあるだけで、Phasing 精度は大幅に減少する。スイッチエラー率が局所的な phasing 性能を評価するのに対して、Phasing 精度は大域的な phasing 性能を評価するために用いた。

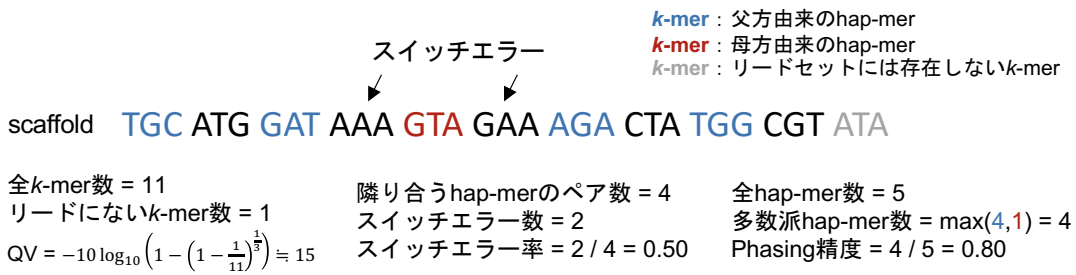

図 26　*k*-mer ベースによる精度評価

k-mer ベースによる精度評価の模式図。*k* = 3 の時の例。実際は *k*-mer は 1 文字ずらしで計算するが、図では簡略化のため 3 文字ずらしで描いている。青色が父方由来の hap-mer、赤色が母方由来の hap-mer、灰色がリードセットには存在しない *k*-mer を表す。

・Hap-mer blob plot

アセンブル結果の phasing 精度を可視化するために、Merqury を用いて hap-mer blob plot を作成した。Hap-mer blob plot とは、各 scaffold にそれぞれの親由来の hap-mer がどれだけ存在するかを散布図でプロットしたものである。各円が一つの scaffold を示しており、円の大きさは scaffold 長を表す。X 軸、Y 軸はそれぞれ scaffold 内の父方、母方由来の hap-mer 数に対応する。Scaffold 内にスイッチエラーがあり、父方由来、母方由来のハプロタイプのモザイクになっていた場合、hap-mer blob plot では対角線付近に配置される。一方、scaffold 内にスイッチエラーがなく片親由来のハプロタイプのみで構成されている場合、hap-mer blob plot では軸に沿って配置される。したがって、hap-mer blob plot では各円(scaffold)が両軸の近くに配置されるほど phasing 精度が高いことを示す。

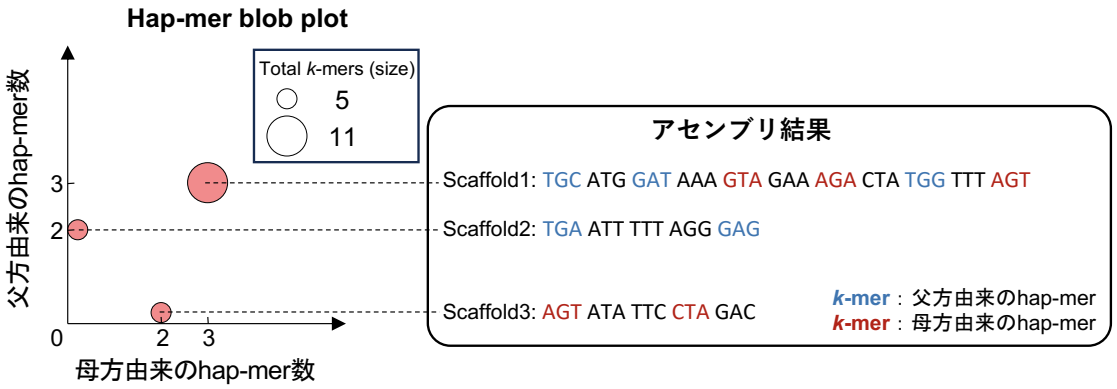

図 27　hap-mer blob plot の説明

Hap-mer blob plot の模式図。$k = 3$ の時の例。実際は k-mer は 1 文字ずらしで計算するが、図では簡略化のため 3 文字ずらしで描いている。青色が父方由来の hap-mer、赤色が母方由来の hap-mer を表す。Scaffold1 は内部にスイッチエラーがあるため父方由来、母方由来のハプロタイプのモザイクになっており、hap-mer blob plot では対角線付近に配置される。Scaffold2、Scaffold3 は、片親由来のハプロタイプのみで構成されており、hap-mer blob plot では軸に沿って配置される。

3.2 ベンチマーク結果

3.2.1 線虫シミュレーションデータによるベンチマーク結果

線虫のシミュレーションデータを作成しベンチマークを行った。

線虫（*Caenorhabditis elegans*）の N2 株と CB4856 株のリードセットを合わせて擬似二倍体のゲノムのリードとして使用した。NCBI データベースよりダウンロードした 2 株のリファレンス配列をシミュレーションデータ作成に使用した。(N2 : GCF_000002985.6、CB4856 : GCA_000975215.1)。Paired-End リードは、ART[77]を用いて coverage 80x のデータを作成した。PacBio の CLR リードは、それぞれの株の実データを DDBJ データベースよりダウンロードした(N2 : DRR142774、CB4856 : DRR142768)。CLR リードに実データを用いた理由は、FALCON-Unzip の入力としてシーケンサーから出力される BAM ファイルが必要だったが、シミュレーションツールでは作成できなかったためである。ダウンロードした CLR リードは、coverage 80x にダウンサンプリングしてベンチマークに使用した。Hi-C リードは、sim3C[78]を用いて coverage 60x のデータを作成した。Sim3C はデフォルトでは環状ゲノムの Hi-C データをシミュレーションするため、liner オプションを使用して線状ゲノムの Hi-C シミュレーションデータを作成した。使用したリードデータの統計値は表 2 に示す。

様々なタイプの入力に対する GreenHill の性能を評価するため、3 種類のタイプの contig を入力として使用した。Paired-haplotype style の入力として Platanus-allee、Pseudo-haplotype style の入力として FALCON-Unzip、Haplotype-ignorant style の入力として Canu によるアセンブル結果を使用した。入力 contig の統計値は表 4 に示す。

このベンチマークは、シミュレーションリードを用いてゲノムを構築しリファレンス配列と比較することで詳細な精度評価を行うために実施した。また、Hi-C のシミュレーションデータを用いることで、topology associating domain (TAD)など局所的な染色体立体構造の影響のない理想的な条件下での基本的な Hi-C Scaffolding、Phasing 機能の性能を調査した。ミスアセンブリ数とスイッチエラー数はリファレンス配列を用いて計算した。

アセンブル結果の統計量を表 5 に示す。FALCON-Unzip を入力とした結果では、FALCON-Unzip＋GreenHill 結果の N50 は約 17.1Mb と最大の値を示しており連続性の高いハプロタイプを構築することができた。一方、FALCON-Unzip ＋ FALCON-Phase ＋ SALSA2 結果の N50 は約 9.5Mb とより断片化していた。精度面では、FALCON-Unzip＋GreenHill 結果のスイッチエラー数が最も少なく、ミスアセンブリ数は SALSA2 に次いで少なかった。Canu を入力とした結果では、Canu＋GreenHill 結果が最も大きな N50 値を示し、ミスアセンブリ数、スイッチエラー数が最も少なかった。さらに、Platanus-allee＋GreenHill 結果は、N50 は約 17.1Mb と連続性が高く、スイッチエラー数は全結果で最

も少ない結果だった。これらの結果から、GreenHill は様々なタイプの入力で連続性の高く高精度なハプロタイプを構築することができ、その汎用性が示唆される。

表 5　線虫データのベンチマーク結果

使用ツール	全長 (bp)	最大長 (bp)	N50 (bp)	ミスアセンブリ数	スイッチエラー数
Platanus-allee + GreenHill	208,814,372	20,808,020	17,065,040	105	1,657
FALCON-Unzip + GreenHill	204,407,441	20,723,971	**17,142,644**	108	**2,121**
FALCON-Unzip + FALCON-Phase + 3D-DNA	226,102,670	22,381,602	16,567,875	286	2,444
FALCON-Unzip + FALCON-Phase + SALSA2	216,393,292	13,113,712	9,455,592	**104**	2,244
Canu + GreenHill	195,748,221	19,835,919	**16,219,706**	**182**	**1,742**
Canu + FALCON-Phase+ 3D-DNA	223,116,127	21,318,404	14,712,650	665	2,165
Canu + FALCON-Phase+SALSA2	209,942,443	15,218,531	6,564,248	315	1,989

全長、N50、ミスアセンブリ数、スイッチエラー数は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された scaffold の合計長を表す。ミスアセンブリ数とスイッチエラー数は、リファレンス配列に基づく方法で計算した。太字の値は、各入力アセンブリに対して最良の結果を示す。

3.2.2 ショウジョウバエデータによるベンチマーク結果

ショウジョウバエのロングリードの実データ、Hi-C リードのシミュレーションデータを使用してベンチマークを行った。

サンプルはキイロショウジョウバエ（*Drosophila melanogaster*）の ISO1 株と A4 株をかけ合わせた雑種第一代(F1)個体を使用した。リファレンス配列は、NCBI データベースよりダウンロードしてベンチマークに使用した(ISO1：GCF_000001215.4、A4：GCA_003401745.1)。CLR リード、HiFi リードは、先行研究[35]で使用されていた実データをダウンロードした(https://obj.umiacs.umd.edu/marbl_publications/hicanu/index.html)。Hi-C リードは、sim3C を用いて linear オプションで coverage 56x のシミュレーションデータを作成した。使用したリードデータの統計値は表 2 に示す。

入力の contig として、HiFi リードについては HiCanu と hifiasm、CLR リードについては Canu によるアセンブル結果を使用した。入力 contig の統計値は表 4 に示す。

このベンチマークは、精度の高いロングリードである HiFi リードでの GreenHill の性能を評価するために行った。CLR リードも使用し、HiFi リードと結果を比較した。ミスアセンブリ数とスイッチエラー数はリファレンス配列を用いて計算した。

アセンブル結果の統計値を表 6 に示す。連続性を比較すると、GreenHill 結果の N50 値は全てのケースで既存ツールの結果を上回り、連続性の高いハプロタイプを構築した。また、精度を比較すると、GreenHill 結果はほぼ全てのケースでミスアセンブリ数、スイッチエラー数が最小であった(スイッチエラー数は HiCanu を入力とした際に SALSA2 より多いが差はほぼない)。全てのケースで GreenHill 結果のミスアセンブリ数は既存ツール結果の約半分であり、GreenHill がより高精度にゲノムを構築できていることを示唆する。これらの結果から、GreenHill は HiFi リード、CLR リードどちらも入力でき汎用性が高いことが示された。

表 6　ショウジョウバエデータのベンチマーク結果

使用ツール	Input for contigs	全長 (bp)	最大長 (bp)	N50 (bp)	ミスアセンブリ数	スイッチエラー数
Canu + GreenHill	CLR	298,393,407	32,550,496	**25,267,681**	**515**	**2,316**
Canu + FALCON-Phase+3D-DNA		377,404,855	14,057,924	637,829	1,095	2,604
Canu + FALCON-Phase+ SALSA2		318,193,525	23,535,552	12,777,832	661	2,409
HiCanu + GreenHill	HiFi	321,609,584	33,217,621	**24,975,482**	**930**	686
HiCanu + FALCON-Phase+3D-DNA		351,304,469	21,680,562	2,482,127	1,631	688
HiCanu + FALCON-Phase+SALSA2		297,968,925	28,602,144	24,843,359	1,435	**646**
Hifiasm + GreenHill	HiFi	307,145,492	27,892,039	**24,570,326**	**742**	**480**
Hifiasm + FALCON-Phase+3D-DNA		371,012,458	33,126,928	1,795,858	2,546	752
Hifiasm + FALCON-Phase+SALSA2		327,819,782	25,140,693	9,592,386	2,330	735
Hifiasm Hi-C mode		308,439,591	26,052,400	21,522,312	1,346	637

全長、N50、ミスアセンブリ数、スイッチエラー数は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された scaffold の合計長を表す。ミスアセンブリ数とスイッチエラー数は、リファレンス配列に基づく方法で計算した。太字の値は、各入力アセンブリに対して最良の結果を示す。

3.2.3 ウシ実データによるベンチマーク結果

ウシの実データを用いてベンチマークを行った。

サンプルには、コブウシ(*Bos indicus*)とウシ(*Bos taurus*)の交雑種を使用した。NCBI データベースより Paired-End リード、CLR リード、Hi-C リードをダウンロードしてベンチマークに使用した(Paired-End : SRR6691721–SRR6691727、SRR6691748、SRR6691951–SRR6691953、SRR6691961、 CLR : SRR8224240–SRR8224250、SRR8695274、SRR6691737、SRR6691756–SRR6691758、 SRR6691760、SRR6691761、SRR6691781、SRR6691805、SRR6691818、SRR6691819、 SRR6691839、SRR6691844、SRR6691846、SRR6691858、SRR6691885 、 SRR6691887–SRR6691898 、 SRR6691900 、 SRR6691916 、 SRR6691919–SRR6691929、SRR6691945、 SRR6691972、SRR6691973、SRR6691976–SRR6691983、SRR8872908–SRR8872920、 Hi-C : SRR6691720)。また、NCBI データベースより両親の Paired-End リードをダウンロードし、評価に使用した(*B. indicus* : SRR6691719、SRR6691880 、 SRR6691881 、 SRR6691906 、 *B. taurus* : SRR6691901–SRR6691903 、SRR6691907)。使用したリードデータの統計値は表 2 に示す。

入力の contig としては、FALCON-Unzip によるアセンブル結果のみを使用した。他のアセンブラとして Canu も試したが、計算が 1 ヶ月以上終わらなかったため入力として使用しなかった。また、Platanus-allee によるアセンブル結果はスイッチエラー率が 6%以上と高く精度が悪かったため使用しなかった。これは、入力として用いた Paired-End リードが NextSeq500 の 2-channel SBS 技術によるもので品質が低かったためと考えられる。一方、FALCON-Unzip の結果は、スイッチエラー率は 0.18%と低く、N50 は 4.6Mb と連続性が高いことから、高品質で下流の解析に適していると考えられる(表 4)。

このサンプルの推定ゲノムサイズは約 3Gb(表 3)と大きく、比較的ゲノムサイズが大きいゲノムの実データでの GreenHill の性能をこのベンチマークでは調査した。

アセンブル結果の統計値を表 7 に示す。FALCON-Unzip + GreenHill 結果は、最大の N50 値 (89.8Mb)を達成し、既存ツールの結果を上回る連続性を示した。さらに、FALCON-Unzip + GreenHill 結果は既存ツールよりも高精度で、QV が最も高く、スイッチエラー率が最も低く、Phasing 精度が最も高かった。

表 7　ウシデータのベンチマーク結果

使用ツール	全長 (bp)	最大長 (bp)	N50 (bp)	QV	スイッチエラー率	Phasing 精度
FALCON-Unzip + GreenHill	5,265,512,220	156,630,926	**89,758,138**	**42.40**	**0.18**	**0.949**
FALCON-Unzip + FALCON-Phase + 3D-DNA	5,649,125,145	156,722,321	77,197,018	41.15	0.23	0.826
FALCON-Unzip + FALCON-Phase + SALSA2	5,479,993,517	139,927,026	68,805,579	41.25	0.22	0.856

全長、N50 は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された scaffold の合計長を表す。QV、スイッチエラー率は Merqury で計算した。Phasing 精度は、scaffold 内の多数派のハプロタイプの割合を表し、この指標の値が高いほど、全域的な phasing の性能が高いことを表す。太字の値は、各入力アセンブリに対して最良の結果を示す。

アセンブル結果の phasing 精度を可視化するため、Merqury を用いて hap-mer blob plot を作成した(図 28)。Hap-mer blob plot では、各 scaffold は円で表され、その大きさは scaffold の長さを示している。X 軸と Y 軸には、それぞれ scaffold 内の母親と父親由来の hap-mer 数が示されている。Phasing 精度が高いほど、もう一方のハプロタイプからの hap-mer の数は少なくなるため、円は軸に近い位置に配置される。FALCON-Unzip + GreenHill 結果の hap-mer blob plot の円は、FALCON-Phase ベースのアプローチよりも軸に近い位置に配置されており、FALCON-Unzip + GreenHill 結果が FALCON-Phase ベースのアプローチよりも phasing 精度が優れていることを示している。

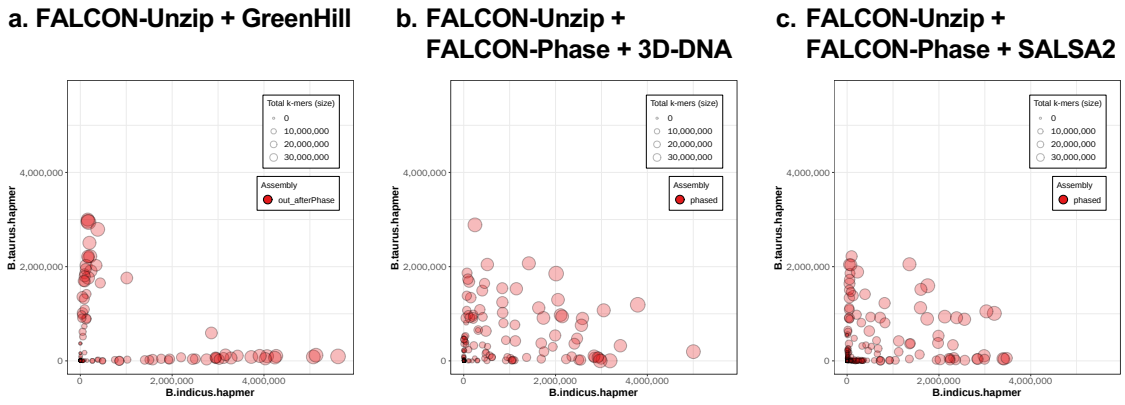

図 28　hap-mer blob plot 結果 (ウシデータ)

染色体レベルでゲノムが構築できているかを確認するため、アライメントドットプロット図の作成を行った。作成には MUMer[79]パッケージの nucmer、delta-filter、mummerplot を使用した。以下に手順を述べる。

i. アセンブル結果の scaffold をリファレンス配列にアライメントした。リファレンス配列には、TrioCanu[51]結果を元に構築したゲノム配列を NCBI データベースよりダウンロードして使用した(GCA_003369685.2、GCA_003369695.2)。Nucmer を用いてアセンブル結果とリファレンス配列のアライメントを行い、アライメント結果を delta-filter によりフィルタリングした。

ii. アセンブル結果の scaffold をリファレンス配列にマッピングし、scaffold のリファレンス配列上の位置(染色体番号、領域)を決定した。マッピングには、minimap2 を使用し、オプションには"-c -k19"を用いた。マッピング結果のうち、同じ向きで同じ染色体にマッピングされる結果を chaining し一つにまとめた。マッピング結果のうちアライメントスコアが最大の位置をその scaffold の位置として決定した。アライメントスコアは、minimap2 の結果の PAF ファイルの AS タグから計算した。

iii. Mummerplot 改良版を使用し、アライメントドットプロットを作成した。iiの scaffold の位置情報とiのアライメント情報を用いて、1 番染色体の scaffolds とリファレンスのアライメントドットプロットを作成した。

1 番染色体のアラインメントドットプロットの結果を図 29 に示す。横軸がリファレンス配列、縦軸がアセンブル結果となっており、両者で相同な領域にドットがプロットされている。ドットの色により、リファレンス配列とアセンブル結果の配列の類似性(Identity)の高さが表されている。また、黒い横点線は、scaffold の切れ目を表している。FALCON-Phase ベースのアプローチでは、2 つのハプロタイプ間で大きなハプロタイプブロックを入れ替える大きなスイッチエラーがあり、FALCON-Unzip + FALCON-Phase + 3D-DNA 結果では逆位のミスアセンブリがあった。一方、FALCON-Unzip + GreenHill 結果では、大きなスイッチエラーやミスアセンブリは見られなかった。また、連続性については、FALCON-Unzip + FALCON-Phase + SALSA2 結果は 1 番染色体の各ハプロタイプが複数の scaffold（> 1 Mb）に分かれており断片化していたが、FALCON-Unzip + GreenHill 結果では、1 番染色体の各ハプロタイプをそれぞれ一つの scaffold (>1Mb)で構築した。この結果から、GreenHill は染色体レベルで繋がったハプロタイプ配列を構築できることが示唆される。

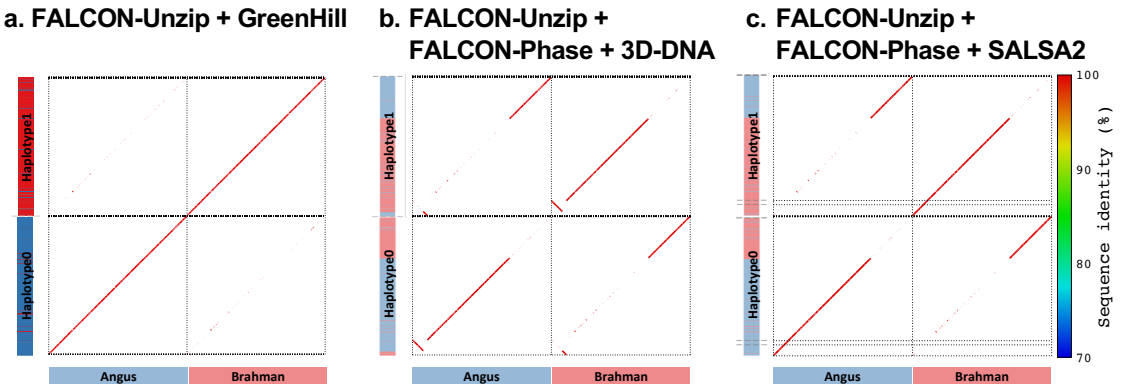

図 29　1 番染色体のドットプロット図 (ウシデータ)

3.2.4 キンカチョウ実データによるベンチマーク結果

キンカチョウの実データを用いてベンチマークを行った。

サンプルは、VGP によりゲノム構築、公開されているキンカチョウ(*Taeniopygia guttata*)のトリオデータ(子供と両親のデータ)を使用した。GenomeArk データベースより、子供の 10X リード、CLR リード、HiFi リード、Hi-C リード、両親の Paired-End リードをダウンロードした(https://vgp.github.io/genomeark/Taeniopygia_guttata)。両親の Paired-End リードは評価にのみ使用した。使用したリードデータの統計値は表 2 に示す。

入力の contig は、CLR リードについては Platanus-allee、FALCON-Unzip、Canu を、HiFi リードについては hifiasm を使用して作成した。Platanus-allee への入力には、Paired-End リードが入手できなかったため 10X リードを Paired-End リードの代わりに用いた。入力 contig の統計値は表 4 に示す。

k-mer 出現頻度分布による解析により、このサンプルの推定ヘテロ接合度は 1.47%(表 3)であり、ヘテロ接合度が高い実データでの性能をこのベンチマークでは検証した。

アセンブル結果の統計量を表 8 に示す。FALCON-Unzip を入力とした結果では、FALCON-Unzip + GreenHill 結果は、FALCON-Unzip + FALCON-Phase + 3D-DNA 結果(N50: 73.4Mb)と比較して、同等の N50 値(70.6Mb)の scaffold を生成した。一方、FALCON-Unzip + FALCON-Phase + SALSA2 結果は、scaffold はかなり断片化していた（N50: 14.2 Mb）。精度については、GreenHill は、QV、スイッチエラー率、Phasing 精度において他のアプローチを上回った。Canu を入力として使用した結果の場合、Canu + GreenHill 結果は Canu + FALCON-Phase + 3D-DNA（74.6 Mb）と同等の N50 値（70.9 Mb）の scaffold を生成し、最高の QV と Phasing 精度を達成した。また、CLR を入力として用いた contig と Hi-C Scaffolding、Phasing ツールの組み合わせでは、Platanus-allee + GreenHill が最も高い Phasing 精度を達成した。入力に使用した contig で結果を比較すると、Canu を用いて得られた contig を入力に使用した結果は、他のツールから得られた結果を入力にした

場合よりもスイッチエラー率が高く、Phasing 精度が低かった。このことは、Canu のようなハプロタイプを無視したスタイルの入力 contig は、phasing において比較的扱いにくいことを示唆している。それにもかかわらず、GreenHill は各入力 contig セットで最高の精度を示し、入力 contig に対する汎用性を裏付けている。Hifiasm を入力として使用した結果は、hifiasm + GreenHill 結果が最大の N50 値(62.8 Mb)を持つ連続性の高い scaffold を生成したのに対し、他のツールはより断片的な scaffold(N50＜12 Mb)を生成した。Hifiasm の Hi-C モードは、QV、スイッチエラー率、Phasing 精度の点で最も正確な結果を生成したが、結果は hifiasm + GreeHill 結果よりも断片化していた。

表 8　キンカチョウデータのベンチマーク結果

使用ツール	Input for contigs	全長 (bp)	最大長 (bp)	N50 (bp)	QV	スイッチエラー率	Phasing 精度
Platanus-allee + GreenHill	10X, CLR	2,309,934,319	152,477,488	61,881,567	35.55	0.57	0.953
FALCON-Unzip + GreenHill	CLR	2,025,894,925	150,748,938	70,617,212	**35.97**	**0.79**	**0.886**
FALCON-Unzip + FALCON-Phase+ 3D-DNA		2,165,277,869	153,652,046	**73,366,558**	35.29	0.87	0.639
FALCON-Unzip + FALCON-Phase+ SALSA2		2,153,682,369	67,065,260	14,220,490	35.35	0.87	0.717
Canu + GreenHill	CLR	1,995,878,168	148,688,728	70,920,789	**35.98**	2.28	**0.849**
Canu + FALCON-Phase + 3D-DNA		2,283,995,575	163,319,697	**74,641,957**	35.57	2.22	0.589
Canu + FALCON-Phase + SALSA2		2,263,303,629	27,304,387	6,544,809	35.61	**2.22**	0.658
Hifiasm + GreenHill	HiFi	2,139,611,083	152,614,629	**62,848,794**	49.40	0.02	0.914
Hifiasm + FALCON-Phase + 3D-DNA		2,667,759,650	77,682,004	1,858,283	49.83	0.02	0.929
Hifiasm + FALCON-Phase + SALSA2		2,352,582,037	64,427,564	11,305,740	49.84	0.02	0.784
Hifiasm Hi-C mode		2,172,318,724	33,557,543	7,896,913	**50.51**	**0.01**	**0.997**

全長、N50 は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された scaffold の合計長を表す。QV、スイッチエラー率は Merqury で計算した。Phasing 精度は、scaffold 内の多数派のハプロタイプの割合を表し、この指標の値が高いほど、全域的な phasing の性能が高いことを表す。太字の値は、各入力アセンブリに対して最良の結果を示す。

アセンブル結果の phasing 精度を可視化するため、Merqury で hap-mer blob plot を作成した(図 30)。FALCON-Phase ベースのアセンブル結果は、hap-mer blob plot で対角線付近に配置される scaffold が多いことから、父方と母方由来のハプロタイプが混ざって構築されている scaffold が多く phasing 精度が悪いことがわかる。また、scaffold 内の父方由来、母方由来の hap-mer 数が少なく原点近くに scaffold が多く配置されていること、円の大きさ(scaffold 長を表す)が比較的小さいものが多いことから、FALCON-Phase ベースのアセンブル結果は連続性が低い傾向があることがわかる。一方、GreenHill ベースのアセンブル結果は、より軸に沿って scaffold が配置されていることから、父方と母方由来のハプロタイプの混同は少なく phasing 精度は既存ツールより高いことがわかる。また、原点から遠く円の大きさが大きい scaffold があることから連続性の高い scaffold を構築できていることがわかる。Hifiasm Hi-C mode の結果は、ほぼ全ての scaffold が軸に沿って配置されていることから、phasing 精度は最も高いことがわかる。しかし、scaffold は断片化しており、原点付近に多くの scaffold が配置され円の大きさも小さい。これらの結果から、GreenHill は他ツールと比較して、連続性、正確性を併せ持つハプロタイプを構築することができることが示唆される。

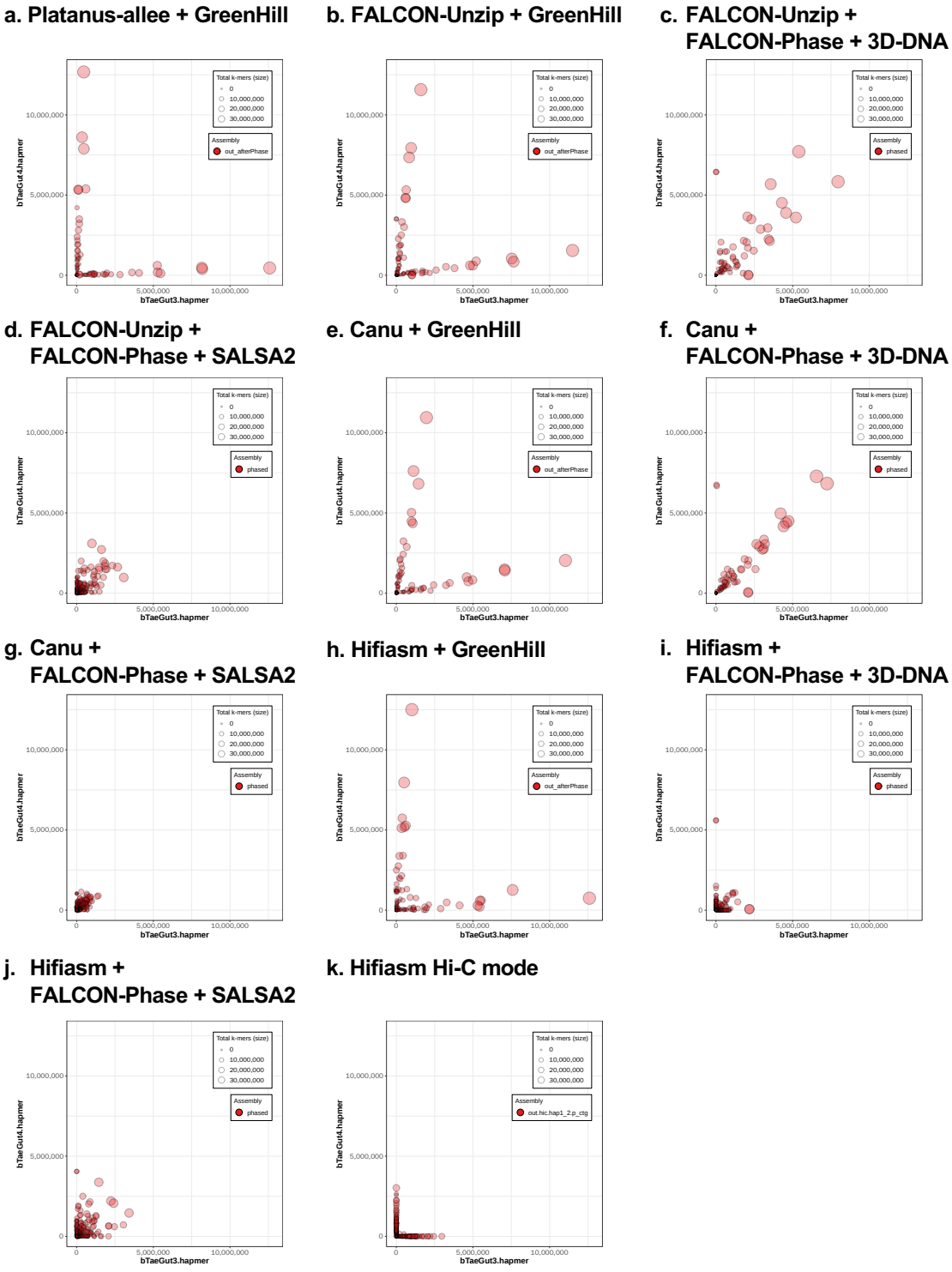

図 30　hap-mer blob plot 結果(キンカチョウデータ)

さらに染色体レベルでの phasing 精度を検証するために、hap-mer を用いて scaffold を由来するハプロタイプに基づいて色分けした。以下に詳しい方法を述べる。

i. 500 kb 以上の scaffold を overlap なしで固定長(100 kb)の断片に切断した。500 kb 未満の短い scaffold はこの解析には使用しなかった。

ii. 各断片がどちらの親由来のハプロタイプ由来かを決定した。まず、Merqury の hap-mer の情報により、各断片内の父方由来、母方由来の hap-mer 数をそれぞれ計算した。そして断片内の hap-mer 数が多い親のハプロタイプを断片のハプロタイプとして決定した。もし、父方由来、母方由来の hap-mer 数が同数の場合、その断片はホモな領域と判定した。各断片を決定されたハプロタイプの情報をもとに色付けした（母方由来は赤色、父方由来は青色、ホモ領域は灰色）。

iii. scaffold をリファレンス配列にマッピングし、scaffold のリファレンス配列上の位置(染色体番号、領域)を決定した。マッピングには、minimap2 を使用し、オプションには"-c -k19"を用いた。マッピング結果のうち、同じ向きで同じ染色体にマッピングされる結果を chaining し一つにまとめた。マッピング結果のうちアライメントスコアが最大の位置をその scaffold の位置として決定した。アライメントスコアは、minimap2 の結果の PAF ファイルの AS タグから計算した。リファレンス配列には、TrioCanu 結果を元に構築したゲノム配列を GenomeArk データベースよりダウンロードして使用した(https://vgp.github.io/genomeark/Taeniopygia_guttata)。

iv. 各断片を、iiiで決定したリファレンス配列上の位置に従い並べ、iiで決定した色で表示した。各 scaffold の境界は縦の黒点線により表示した。

3 番染色体のハプロタイプ構造を図 31 に示す。FALCON-Phase ベースのアセンブル結果は、途中で父方由来（青色）と母方由来（赤色）のハプロタイプが入れ替わっており、大きなスイッチエラーが起きていることが観察された。また、SALSA2 を使用した結果や hifiasm + 3D-DNA の結果では、縦の黒点線が複数本あることから、3 番染色体が複数の scaffold に断片化していることがわかる。一方、GreenHill ベースのアセンブル結果は、大きなスイッチエラーはほぼなく、高精度に 3 番染色体のハプロタイプを構築することができた。また、hifiasm + GreenHill 以外は、GreenHill の結果では縦の黒点線がなく、3 番染色体の各ハプロタイプを単一の scaffold (>1Mb)として構築することができた。Hifiasm Hi-C mode の結果は、大きなスイッチエラーはほぼなく高精度にハプロタイプを構築することができているが、断片化しており 3 番染色体の各ハプロタイプが複数の scaffold に分かれてしまっていた。これらの結果から、GreenHill は 3 番染色体のハプロタイプ配列を高精度に染色体レベルで構築できたと言える。

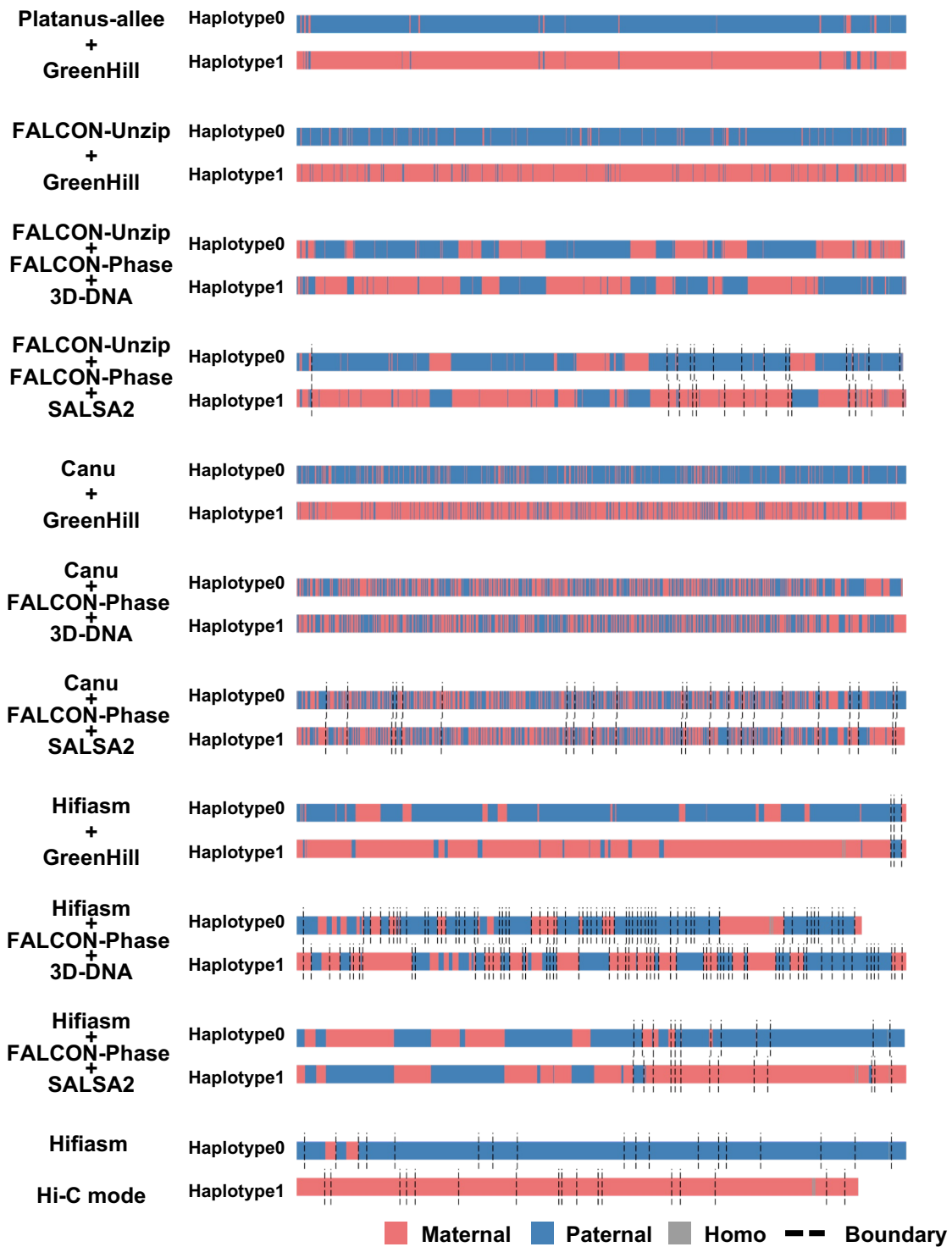

図 31　３番染色体のハプロタイプ構造 (キンカチョウデータ)

3.2.5 他生物種データによるベンチマーク結果

GreenHill の生物種に対する汎用性を調べるために、セキセイインコ、クロサイ、コチョウザメの実データによるベンチマークを行った。

サンプルは、VGP によりゲノム構築、公開されているセキセイインコ(*Melopsittacus undulatus*)、クロサイ（*Diceros bicornis*）、コチョウザメ（*Acipenser ruthenus*）のトリオデータ(子供と両親のデータ)を使用した。GenomeArk データベースより、子供の CLR リードまたは HiFi リード、Hi-C リード、両親の Paired-End リードをダウンロードした(セキセイインコ：https://vgp.github.io/genomeark/Melopsittacus_undulatus、クロサイ：https://vgp.github.io/genomeark/Diceros_bicornis、コチョウザメ：https://vgp.github.io/genomeark/Acipenser_ruthenus)。両親の Paired-End リードは評価にのみ使用した。使用したリードデータの統計値は表 2 に示す。

入力の contig には、CLR リードについては FALCON-Unzip によるアセンブル結果、HiFi リードについて hifiasm によるアセンブル結果を使用した。入力 contig の統計値は表 4 に示す。

k-mer 出現頻度分布による解析によりヘテロ接合度はそれぞれセキセイインコ：1.04%、クロサイ：0.21%、コチョウザメ：0.58%で、ゲノムサイズはそれぞれセキセイインコ：1.12Gb、クロサイ：2.98Gb、コチョウザメ：1.76Gb と推定された(表 3)。ヘテロ接合度やゲノムサイズが異なる様々な生物種でベンチマークを行い、生物種に対する汎用性を検証した。

アセンブル結果の統計値を表 9 に示す。GreenHill は、他のツールと比較して、N50 値が同等（セキセイインコ：88.1Mb）またはそれ以上（クロサイ：52.2Mb、コチョウザメ：25.8Mb）の scaffold を生成した。Hifiasm Hi-C mode は最も正確な結果を生成したが、Hi-C Scaffolding 機能がないため、GreenHill よりも N50 値は低く断片化していた。FALCON-Phase ベースの方法と比較すると、FALCON-Phase ベースの結果の N50 が非常に低い(< 4Mb)場合を除き、GreenHill は、スイッチエラー率、Phasing 精度の点で FALCON-Phase ベースの方法より正確なハプロタイプ配列を構築した。

表 9　他生物種データのベンチマーク結果

生物種	使用ツール	Input for contigs	全長 (bp)	最大長 (bp)	N50 (bp)	QV	スイッチエラー率	Phasing 精度
セキセイインコ	FALCON-Unzip + GreenHill	CLR	2,273,118,342	140,405,933	88,135,996	**40.78**	**0.28**	**0.896**
	FALCON-Unzip + FALCON-Phase + 3D-DNA		2,620,388,952	159,155,647	**90,074,506**	39.20	0.36	0.701
	FALCON-Unzip+ FALCON-Phase + SALSA2		2,437,415,698	95,254,517	34,574,245	39.31	0.34	0.741
クロサイ	Hifiasm + GreenHill	HiFi	5,325,705,542	101,752,101	**52,259,952**	58.44	0.05	0.982
	Hifiasm + FALCON-Phase + 3D-DNA		6,523,963,142	136,733,613	48,162,421	55.16	0.09	0.615
	Hifiasm + FALCON-Phase + SALSA2		6,206,904,826	40,586,421	3,069,332	55.53	**0.05**	0.785
	Hifiasm Hi-C mode		6,047,793,056	94,248,160	30,503,132	**67.13**	0.06	**0.995**
コチョウザメ	Hifiasm + GreenHill	HiFi	3,712,891,508	83,435,872	**25,813,405**	58.82	0.03	0.849
	Hifiasm + FALCON-Phase + 3D-DNA		3,882,750,516	14,958,612	1,019,399	54.98	0.04	0.889
	Hifiasm + FALCON-Phase + SALSA2		3,474,135,770	47,228,216	7,128,235	55.15	0.04	0.751
	Hifiasm Hi-C mode		3,748,196,843	55,271,118	9,593,284	**60.99**	**0.02**	**0.975**

全長、N50 は、長さ≧500bp の配列について計算した。全長は各ツールにより生成された scaffold の合計長を表す。QV、スイッチエラー率は Merqury で計算した。Phasing 精度は、scaffold 内の多数派のハプロタイプの割合を表し、この指標の値が高いほど、全域的な phasing の性能が高いことを表す。太字の値は、各入力アセンブリに対して最良の結果を示す。

さらに phasing 精度を hap-mer blob plot により可視化して評価した(図 32、図 33、図 34)。GreenHill ベースのアセンブル結果の phasing 精度は、hifiasm Hi-C mode のアセンブル結果よりわずかに低いが、FALCON-Phase ベースのアセンブル結果より高いことが観察された。これらの結果は、GreenHill が様々な生物種で使用できる汎用性の高さを示している。

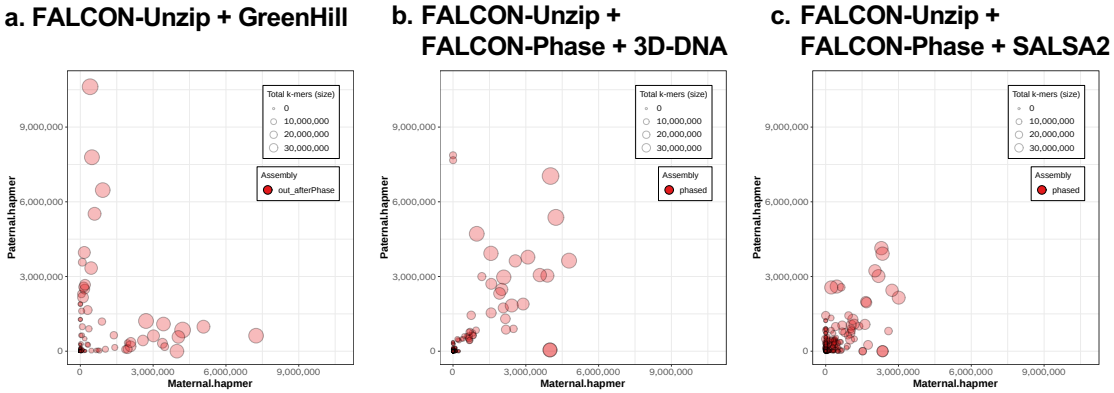

図 32　hap-mer blob plot 結果 (セキセイインコデータ)

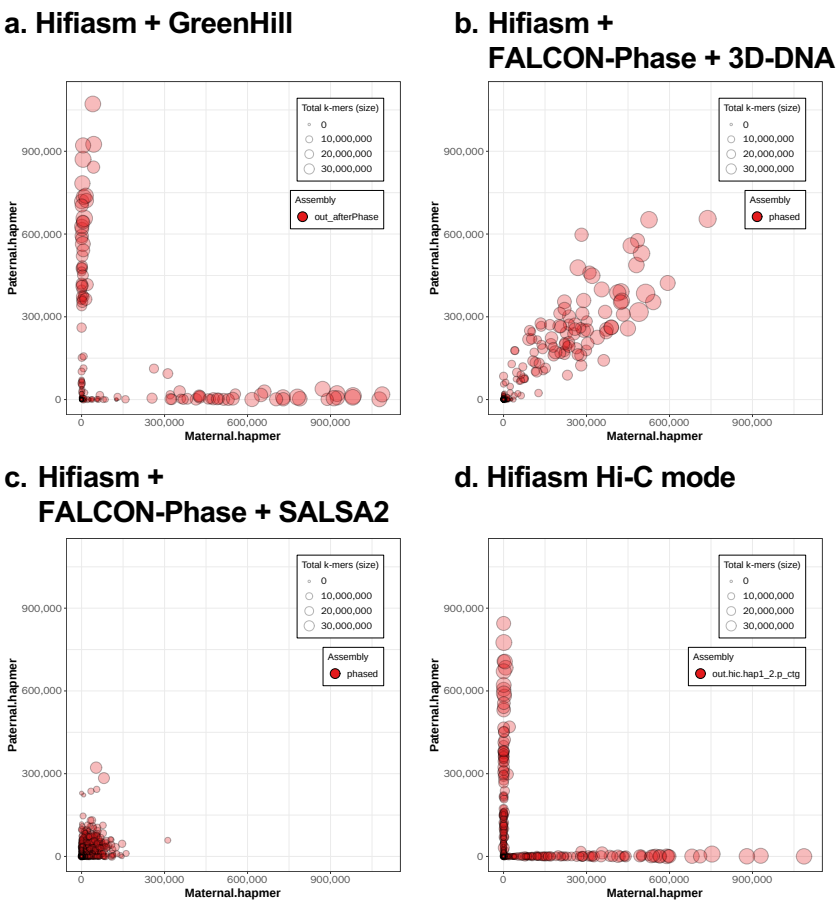

図 33　hap-mer blob plot 結果(クロサイデータ)

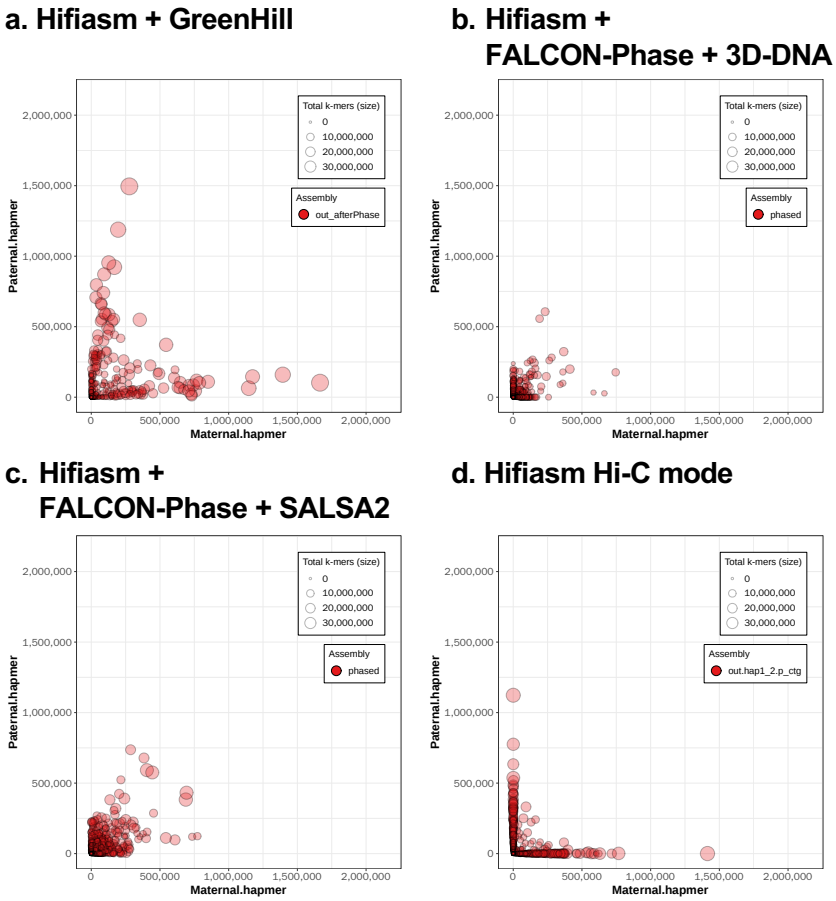

図 34　hap-mer blob plot 結果(コチョウザメデータ)

3.2.6 ノックアウトテスト

GreenHill の特徴的な機能が実際に連続性、精度等に寄与しているかを調べるため、それぞれの機能を無効化(ノックアウト、KO)したベンチマークを行った。ここで対象とした機能は、ロングリードを Hi-C と同時に使用する機能と、Hi-C のコンタクトマップによるミス検出機能である。ミスアセンブリ数はリファレンス配列にアライメントすることで計算した。ウシ、キンカチョウは TrioCanu ベースで構築したゲノムをリファレンス配列として使用した。また、スイッチエラー数、Phasing 精度は両親の Paired-End リードを用いて Merqury で計算した。

ノックアウトテストの結果を表 10 に示す。ロングリードを Hi-C と同時に使用する機能を無効化(simultaneous LR-use KO)すると、ほぼすべての結果で N50 値が下がり、ミスアセンブリ数が増加した。この結果から、ロングリードを Hi-C と同時に使用する機能は、正しく scaffold を接続するのに寄与していると考えられる。Hi-C のコンタクトマップによるミス検出機能を無効化(Hi-C-based correction KO)すると、ほぼすべての結果

でN50値は増加し、Phasing精度は低下した。N50値が増加したのは、Hi-Cのコンタクトマップによるミス検出機能を無効化したことで異なる染色体のscaffold同士を接続してしまったのが原因と考えられる。そのため、アセンブリ結果の最大長がリファレンス配列の最大の染色体の長さより長くなっている。この結果から、Hi-C のコンタクトマップによるミス検出機能が異なる染色体同士を接続するのを防いでいることが示唆される。

表 10　ノックアウトテストの結果

生物種	入力	ノックアウト(KO)	最大長 (bp)	N50 (bp)	ミスアセンブリ数	スイッチエラー率	Phasing精度
線虫	Platanus-allee	default	20,808,020	17,065,040	105	1.16	0.867
		simultaneous LR-use KO	18,821,896	13,775,982	170	1.05	0.881
		Hi-C-based correction KO	53,974,611	53,973,479	90	1.17	0.825
	FALCON-Unzip	default	20,723,971	17,142,644	108	2.15	0.801
		simultaneous LR-use KO	14,847,431	8,543,164	338	2.11	0.877
		Hi-C-based correction KO	35,694,521	17,454,434	135	2.16	0.799
	Canu	default	19,835,919	16,219,706	182	3.04	0.883
		simultaneous LR-use KO	3,504,707	209,411	272	3.31	0.959
		Hi-C-based correction KO	65,498,193	65,498,172	209	3.01	0.887
ショウジョウバエ	Canu	default	32,550,496	25,267,681	515	4.42	0.803
		simultaneous LR-use KO	30,328,540	23,320,938	652	4.48	0.828
		Hi-C-based correction KO	52,685,516	32,655,313	543	4.45	0.809
	HiCanu	default	33,217,621	24,975,482	930	0.03	0.930
		simultaneous LR-use KO	27,085,171	22,450,516	944	0.03	0.927
		Hi-C-based correction KO	111,531,930	111,531,912	931	0.03	0.917
	Hifiasm	default	27,892,039	24,570,326	742	0.03	0.920
		simultaneous LR-use KO	27,007,628	22,427,718	738	0.03	0.923
		Hi-C-based correction KO	49,556,982	23,155,838	766	0.03	0.928
ウシ	FALCON-Unzip	default	156,630,926	89,758,138	7,570	0.18	0.949
		simultaneous LR-use KO	146,308,271	65,105,278	9,863	0.17	0.971
		Hi-C-based correction KO	242,373,730	103,567,514	7,697	0.18	0.907

生物種	入力	ノックアウト(KO)	最大長 (bp)	N50 (bp)	ミスアセンブリ数	スイッチエラー率	Phasing 精度
キンカチョウ	Platanus-allee	default	152,477,488	61,881,567	1,332	0.57	0.953
		simultaneous LR-use KO	88,255,425	33,070,620	2,096	0.50	0.957
		Hi-C-based correction KO	289,099,635	93,668,539	1,285	0.57	0.782
	FALCON-Unzip	default	150,748,938	70,617,212	1,543	0.79	0.886
		simultaneous LR-use KO	131,927,695	54,233,580	2,959	0.79	0.912
		Hi-C-based correction KO	393,590,674	222,441,212	1,615	0.78	0.669
	Canu	default	148,688,728	70,920,789	2,741	2.28	0.849
		simultaneous LR-use KO	92,763,168	14,409,374	3,972	2.34	0.900
		Hi-C-based correction KO	453,243,742	262,346,933	2,921	2.29	0.614

N50 は、長さ≧500bp の配列について計算した。ミスアセンブリ数は、リファレンス配列に基づく方法で計算した。ウシ、キンカチョウは TrioCanu ベースの結果をリファレンス配列として用いた。スイッチエラー率は Merqury で計算した。Phasing 精度は、scaffold 内の多数派のハプロタイプの割合を表し、この指標の値が高いほど、全域的な phasing の性能が高いことを表す。太字の値は、各入力アセンブリに対して最良の結果を示す。

3.2.7 実行時間、メモリ使用量についてのベンチマーク結果

３つのツールの実行時間とメモリ使用量をすべてのベンチマークで評価した。GreenHill は、入力 contig のヘテロ領域の対応づけ、Hi-C リードのマッピング、Hi-C Scaffolding、Hi-C Phasing をすべてツール内で行う。そのため、既存ツールの実行時間は、Purge_dups、Hi-C リードのマッピング(Juicer または Arima mapping pipeline)、Hi-C Scaffolding (3D-DNA または SALSA2)、Hi-C Phasing (FALCON-Phase)の実行時間を加算して計算した。ツールの CPU 時間、実時間、最大メモリ使用量は、Intel(R) Xeon(R) Gold 6342 CPU（2.80GHz クロック、デュアル 24 コア）と 512GB の RAM を搭載したコンピュータ上で GNU time コマンドを使用して測定した。各プロセスのスレッド数は、設定可能であれば 48 とした。

実行時間とメモリ使用量の測定結果を表 11 に示す。GreenHill の実行時間は他のツールと同等かそれ以下だった。ゲノムサイズが小さいデータ（<500Mb、線虫、ショウジョウバエ）では約 1 時間以内、ゲノムサイズが大きいデータ（>2Gb、クロサイ）では 2 日以内に GreenHill はハプロタイプを構築した。FALCON-Phase + 3D-DNA、FALCON-Phase + SALSA2 は、特にゲノムサイズが大きい場合に時間がかなりかかった（クロサイのデータでは約 10 日以上)。これは、Hi-C Scaffolding ツール、Hi-C Phasing ツールを実行するたびに Hi-C リードのマッピングを行うためと考えられる。GreenHill の最大メモリ使用量は、他ツールよりも多かったが、ゲノムサイズが大きいデータ(>2Gb、クロサイ)でも 206Gb 以下だった。

表 11　実行時間、メモリ使用量

生物種	入力 contig	使用ツール	CPU time(h)	Real time (h)	Max memory (GB)
線虫	Platanus-allee	GreenHill	13.81	0.53	23.54
	FALCON-Unzip	GreenHill	16.67	**0.74**	23.18
		FALCON-Phase + 3D-DNA	**10.01**	1.63	**17.56**
		FALCON-Phase + SALSA2	10.90	2.03	18.48
	Canu	GreenHill	20.90	**1.03**	24.21
		FALCON-Phase + 3D-DNA	**9.83**	1.50	**17.55**
		FALCON-Phase + SALSA2	10.90	2.08	18.55
ショウジョウバエ	Canu	GreenHill	**22.73**	**1.02**	35.41
		FALCON-Phase + 3D-DNA	26.87	2.94	**18.24**
		FALCON-Phase + SALSA2	27.41	3.29	18.80
	HiCanu	GreenHill	**9.20**	**0.44**	44.94
		FALCON-Phase + 3D-DNA	26.90	3.68	**18.25**
		FALCON-Phase + SALSA2	27.25	3.52	19.32
	Hifiasm	GreenHill	**11.49**	**0.45**	37.04
		FALCON-Phase + 3D-DNA	26.84	3.26	**18.17**
		FALCON-Phase + SALSA2	28.03	3.50	19.03
ウシ	FALCON-Unzip	GreenHill	337.00	15.36	192.03
		FALCON-Phase + 3D-DNA*	309.37	18.72	**96.36**
		FALCON-Phase + SALSA2*	**285.16**	**13.73**	97.10

生物種	入力 contig	使用ツール	CPU time(h)	Real time (h)	Max memory (GB)
キンカチョウ	Platanus-allee	GreenHill	323.32	18.04	108.19
	FALCON-Unzip	GreenHill	296.51	19.41	92.06
		FALCON-Phase + 3D-DNA*	**123.25**	**16.37**	76.32
		FALCON-Phase + SALSA2*	127.30	20.50	**39.46**
	Canu	GreenHill	446.53	26.00	101.89
		FALCON-Phase + 3D-DNA	**201.05**	**22.49**	154.58
		FALCON-Phase + SALSA2	205.86	23.77	**37.09**
	Hifiasm	GreenHill	**102.24**	**7.45**	89.60
		FALCON-Phase + 3D-DNA	229.99	25.52	156.29
		FALCON-Phase + SALSA2	209.47	24.44	**50.37**
セキセイインコ	FALCON-Unzip	GreenHill	255.34	31.30	90.52
		FALCON-Phase + 3D-DNA	240.95	41.59	108.85
		FALCON-Phase + SALSA2	**201.35**	**27.58**	**50.53**
クロサイ	Hifiasm	GreenHill	**579.32**	**26.80**	206.37
		FALCON-Phase + 3D-DNA	9,535.56	277.73	98.92
		FALCON-Phase + SALSA2	8,543.02	238.56	**65.96**
コチョウザメ	Hifiasm	GreenHill	**569.34**	**29.55**	128.75
		FALCON-Phase + 3D-DNA	1,300.94	71.28	80.00
		FALCON-Phase + SALSA2	1,242.32	79.40	**65.96**

実行時間とメモリ使用量の概要。太字は各入力アセンブリに対して最良の結果を示す。
※ウシとキンカチョウの FALCON-Unzip + FALCON-Phase + Hi-C Scaffolding tools の結果は、FALCON-Unzip + FALCON-Phase の結果をダウンロードして使用しているため、実行時間に FALCON-Phase round 1 の時間は含まれていない。

3.3 考察

本ベンチマークでは、様々なタイプの入力 contig と多くの生物種のデータを用いて、GreenHill の性能と汎用性を検証した。第一のベンチマーク(線虫)、第二のベンチマーク(ショウジョウバエ)では、Hi-C のシミュレーションデータを用いてテストを行い、染色体の局所的な立体構造の影響がない理想的な条件下で、GreenHill の基本性能を確認した。Hi-C 法は、本来染色体の立体構造解析のために開発された手法であり、立体的に距離が近いゲノム間で架橋数が多くなるよう設計されている。Hi-C Scaffolding では、その性質を利用し、ゲノム配列上の 1 次元の距離が近ければ立体的距離も近いため Hi-C の架橋数が多いことを前提として行われている。しかし、実際は TAD や A/B コンパートメントなど染色体の局所的な立体構造により、配列上の 1 次元の距離と立体的距離は必ずしも比例しない。そのため、Hi-C の実データを用いた Hi-C Scaffolding は、染色体の局所的な立体構造の影響を受けより難しいと考えられる。第一、第二のベンチマークでは、そのような影響のない理想的なシミュレーションデータで基本的な性能を評価しており、線虫のデータでは CLR、ショウジョウバエのデータでは HiFi を使用した際の基本的な性能を確認した。線虫、ショウジョウバエのデータの両方で、GreenHill は既存ツールより N50 値が大きく、ミスアセンブリ数、スイッチエラー数が少ないハプロタイプ配列を作成し、GreenHill が高精度で連続性の高いハプロタイプ配列を構築できる高い性能があることが示された。

第三のベンチマーク(ウシ)、第四のベンチマーク(キンカチョウ)、第五のベンチマーク(セキセイインコ、クロサイ、コチョウザメ)では、Hi-C の実データを用いてテストを行い、実データでの GreenHill の実際の性能を評価した。サンプルとして、両親と子供のデータのあるトリオデータを使用し、子供のデータのみでゲノム構築し、両親データを用いて phasing 性能を評価した。これらのベンチマークテストでは、ゲノムサイズが大きいゲノム（ウシ、クロサイ）やヘテロ接合度が高いゲノム（キンカチョウ)、様々な生物種（哺乳類、鳥類、魚類）を用いたテストを行うことで、サンプルに対する汎用性を検証した。GreenHill はすべてのサンプルで、既存ツールと同等または上回る連続性及び 0.849～0.982 の高い Phasing 精度を示し、高い汎用性が示唆された。精度の面では、GreenHill は、hifiasm Hi-C mode に少し劣るが、hifiasm Hi-C mode は Hi-C による scaffolding 性能がないため連続性が低いことを考慮すると、GreenHill は既存ツールより高い連続性、精度でハプロタイプを構築できたと言える。さらには、GreenHill は、1 本の染色体のほぼ全長を phasing した scaffold を構築しており、全染色体の完全長のハプロタイプを *de novo* でアセンブルするという目標に近い結果を達成することができた。

本ベンチマークでは、0.21～1.47%まで様々なヘテロ接合度の生物種のデータを用いて GreenHill の性能を評価した(表 3)。ヘテロ接合度は、ゲノムアセンブリの品質に大きな影響を与えることが想定される。ヘテロ接合度が高い場合、アセンブルグラフが複雑

になることで断片化やミスアセンブリを引き起こしやすい。一方、ヘテロ接合度が低い場合、ハプロタイプ間で違いのあるヘテロな領域が少なく、ヘテロ領域間を架橋するリードペアが少なくなるため、phasing が困難になる。サンプルのヘテロ接合度により、N50、スイッチエラー率、Phasing 精度がどのように変化するかを図 35、図 36 に示す。入力 contig の条件を同じにするため、FALCON-Unzip、hifiasm を入力として使用したベンチマーク結果を図にまとめた。

まず、N50 については、ヘテロ接合度が高くなると、FALCON-Phase＋3D-DNA の結果では hifiasm が入力の場合、N50 が大幅に下がり断片化した。一方、GreenHill の結果ではヘテロ接合度によらず、既存ツールと同等または大きい N50 値を示しており、連続性の高いハプロタイプを構築することができていることが確認できる。次にスイッチエラー率については、FALCON-Unzip のアセンブル結果が入力の場合、ツール間であまり差はなく入力の contig とほぼ同じエラー率だった。これは、FALCON-Unzip はエラーの多い CLR リードを元に phasing を行うためスイッチエラー率が高く、最終結果までスイッチエラーが残ってしまったためと考えられる。Hifiasm のアセンブル結果が入力の場合、ヘテロ接合度が低いほどスイッチエラー率が高い傾向があった。GreenHill はヘテロ接合度が低い場合(クロサイ：0.21)でも、スイッチエラー率は FALCON-Phase＋SALSA2 の結果に次いで少なく、高精度に phasing をすることができていた。Phasing 精度については、ヘテロ接合度とあまり相関は見られなかった。FALCON-Phase＋3D-DNA の結果は、ヘテロ接合度が高くなると Phasing 精度が上がっているが、これは N50 が低下して断片化していることが理由と考えられる。GreenHill はヘテロ接合度によらず、FALCON-Unzip のアセンブル結果の入力でも 0.80 以上の Phasing 精度でハプロタイプを構築していた。これらの結果から、GreenHill のヘテロ接合度に対する汎用性の高さが示唆されている。

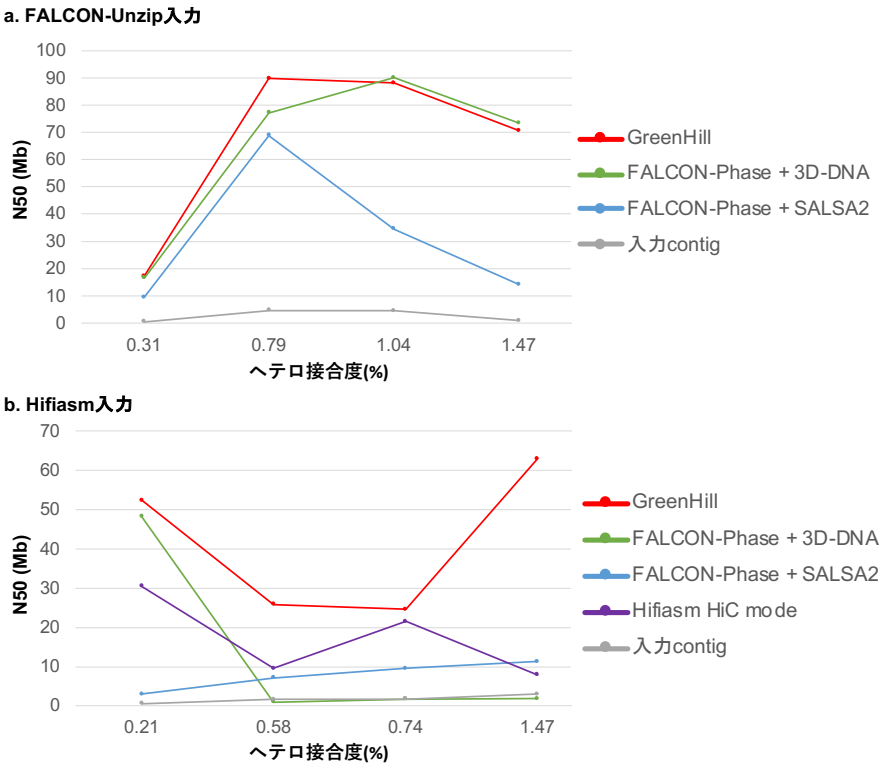

図 35　ヘテロ接合度と連続性の関係

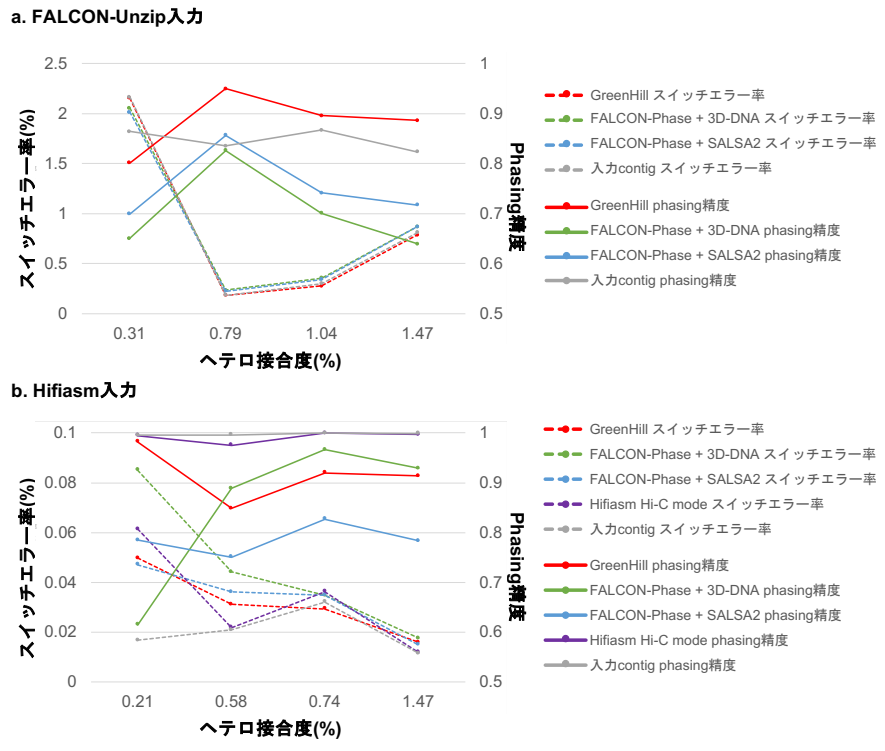

図 36　ヘテロ接合度と精度の関係

入力データについては、GreenHill は様々なタイプのアセンブラから出力される contig を使用でき、完全または部分的に phasing された contig も使用することが可能である。さらに、contig が primary contig と alternative contig に分類されず取り扱いが困難だった Canu などの Haplotype-ignorant style の contig でもハプロタイプを構築することができており、GreenHill は入力に対して高い汎用性を持っていることが確認できる。入力とするロングリードについても、エラー率が高い(10‐15%)PacBio CLR リードとエラー率が低い(<1%)HiFi リード[18]の両方をベンチマークに使用しており、CLR＋Hi-C データ、HiFi＋Hi-C データを使用したベンチマークの結果、GreenHill は CLR リード、HiFi リードのどちらでも良好な結果を示した。HiFi リードは、CLR リードよりも高精度のため phasing に有効と考えられる。しかし HiFi のライブラリ作成にはより多くの DNA 量が必要という課題もある。CLR リードの DNA 抽出プロトコルでは、すべての長い断片(例えば>15 kb)を利用できるのに対して、HiFi リードの DNA 抽出プロトコルでは、限られた範囲の長さの断片(例えば 15‐18 kb)しか利用できない[18]。したがって、HiFi リードは、より多くの DNA 量が必要で DNA 抽出が難しいサンプルには適用しにくい可能性があり、CLR の方が多くのサンプルに適用できると言える。本ベンチマーク結果は、エラー率が高くより難易度の高い CLR が入力のケースでも、GreenHill は高い性能を示しており、様々な二倍体生物のゲノム構築への実用性が示唆される。

GreenHill のアルゴリズムに関してユニークな機能としては、ロングリードと Hi-C を同時に使用する機能や、分散ベースの閾値選択[62]を用いた Hi-C コンタクトマップによるエラー修正機能が挙げられる。ロングリードと Hi-C の両方を使用する既存のパイプライン([56]、[59])やプロトコル([63])も存在するが、それぞれのリードを別々に使用しており、同時に考慮することはない。GreenHill は、Hi-C の全域的な情報とロングリードの局所的な情報を同時に考慮することで精度良くハプロタイプ配列を構築することが可能である。また、Hi-C コンタクトマップによるエラー修正も、既存ツール（[22]、[80]）で行われているが、それらのツールでは、あらかじめ設定された閾値を用いて実行されており、その値はサンプルやゲノム領域によらず同じである。そのため、Hi-C の coverage が低い場合やリピート領域が多く Hi-C の mapping 率が低い領域などを誤ってミスと判定し scaffold を切断するケースが存在する。GreenHill は、分散ベースの閾値選択によりサンプル、領域ごとに最適な閾値を決定するため、誤ってミスと判定しにくくなり、連続性高くハプロタイプ配列を構築することができる。これらの機能を無効化したノックアウトテストでも、これらのユニークな機能の有効性を確認することができた。

第 4 章　総括

本研究は、Hi-C 法を用いて染色体レベルのハプロタイプゲノム配列を構築するツールの開発を目的として行われ、Hi-C データを用いて scaffolding、phasing を行い染色体レベルのハプロタイプ配列を構築するツール GreenHill の新規開発に成功した。GreenHill の性能を評価するため、様々な生物種、ヘテロ接合度、入力 contig、入力ロングリードのデータを用いてベンチマークを行い、GreenHill がいずれのサンプルでも優れた性能を発揮し、高精度で連続性の高いハプロタイプを構築できることを示した。これらの結果から、GreenHill は高い汎用性を持ち、様々な二倍体生物のゲノムアセンブリプロジェクトに使用できると考えられる。

多数の生物種の全ゲノム配列解読を目標とした大規模ゲノムプロジェクトは 2023 年時点で複数進行している。昆虫 5000 種のゲノム配列解読を目指す i5K[81]、鳥類 10500 種のゲノム配列解読を目指す Bird 10K[82]、全脊椎動物約 7 万種のゲノム配列解読を目指す Vertebrate Genome Project(VGP)[63]、地球上の全真核生物 150 万種のゲノム配列解読を目指す Earth BioGenome Project(EBP)[83]などが挙げられる。これらの計画では、多くの種の高品質な染色体レベルのゲノム構築を目標としており、そのためには高精度な染色体レベルのゲノム構築の自動化、効率化が必要とされる。しかし、これらのプロジェクトで使用されている既存の Hi-C Scaffolding ツールの結果には、異なる染色体やゲノム領域間を繋ぐミスが残っており、Juicebox[84]などのツールを用いて可視化した Hi-C コンタクトマップを目視で確認し、ミスアセンブリを修正するマニュアルキュレーションが必要なケースが多い。GreenHill は独自の Hi-C コンタクトマップによるミス修正機能があるため、異なる染色体やゲノム領域間を繋ぐミスを減らすことができ、このようなマニュアルキュレーションの負担を軽減できることが見込まれる。ゲノム構築に GreenHill が活用されることにより、より多くの生物種の染色体レベルのゲノムをより簡便に構築できるようになることが期待される。

染色体レベルのハプロタイプゲノムリファレンスは多くの生物学分野で、リソースとして重要である[50]。しかし、既存の *de* novo ハプロタイプ構築手法は、精度や連続性が低かったり、トリオデータや近縁種のリファレンス配列が必要であったりと制限があった。GreenHill は、リファレンスゲノムやトリオデータなどの追加情報なしで、Hi-C データを用いて高精度な染色体レベルのハプロタイプを構築できる。これにより、非モデル生物を含む様々な生物の染色体レベルのハプロタイプ配列が構築できるようになれば、二倍体ゲノムの完全な情報を含むリソースとして様々な分野での活用が期待される。

本研究で主に扱った CLR リード、HiFi リードの他に注目すべきロングリード技術として、Oxford Nanopore Technology(ONT リード)がある。本研究では、ONT リードを用いたベンチマークは行っておらず GreenHill が ONT リード入力に対して有効かは判断できないが、GreenHill はエラーの多い CLR リードに対応しているため、同じくエラー

の多いロングリードである ONT リードに対してもある程度有効であると考えられる。また、新規な技術として、最大で数 Mb の長いリードが得られる ONT ultra-long read(UL)[85]がある。UL リードは、エラー率は高いがリード長が長いためリピート領域を超えてゲノムを構築しやすいという利点がある。例えば、UL リードを使用することで、ヒトの Y 染色体のセントロメアの構築[86]や X 染色体の染色体の端から端まで(Telomere-to-Telomere、T2T)の構築[87]が可能になった。さらに、HiFi リードと UL リードを組み合わせることでヒトの全染色体を T2T で構築[88]を可能にした。UL リードは、コストが高くまだ多くの生物種のゲノムアセンブリには使用されていないが、今後注目すべき技術の一つと言える。

最後に本研究の課題と今後の展望について述べる。課題としては以下の二つが挙げられる。一つ目は、GreenHill は入力 contig のスイッチエラーを検出、修正する機能がないため、GreenHill が構築したハプロタイプのスイッチエラー率は、入力 contig となるデータを構築したアセンブラの精度に依存する点である。リードを contig にマッピングしてスイッチエラーを検出し修正するなどの対策が必要であるが、この問題は今後 HiFi リードを入力とした contig assembly の精度が改善するにしたがい解消されると考えられる。二つ目は、GreenHill は二倍体ゲノム用に設計されているため、多倍体ゲノムを扱うことができない点である。多くの高等真核生物は二倍体であるが、植物など多倍体の生物も存在する。今後の研究では、ハプロタイプの対応付けを 1 対 1 ではなく複数で対応づけられるようにするなど GreenHill の機能を多倍体用に拡張する必要があると考えられる。

参考文献

1. Maxam AM, Gilbert W. A new method for sequencing DNA. Proc Natl Acad Sci USA. 1977;74:560–4.

2. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci USA. 1977;74:5463–7.

3. Prober JM, Trainor GL, Dam RJ, Hobbs FW, Robertson CW, Zagursky RJ, et al. A System for Rapid DNA Sequencing with Fluorescent Chain-Terminating Dideoxynucleotides. Science. 1987;238:336–41.

4. Smith LM, Sanders JZ, Kaiser RJ, Hughes P, Dodd C, Connell CR, et al. Fluorescence detection in automated DNA sequence analysis. Nature. 1986;321:674–9.

5. Ansorge W, Sproat BS, Stegemann J, Schwager C. A non-radioactive automated method for DNA sequence determination. Journal of Biochemical and Biophysical Methods. 1986;13:315–23.

6. Swerdlow H, Gesteland R. Capillary gel electrophoresis for rapid, high resolution DNA sequencing. Nucleic Acids Res. 1990;18:1415–9.

7. Drossman Howard, Luckey JA, Kostichka AJ, D'Cunha Jonathan, Smith LM. High-speed separations of DNA sequencing reactions by capillary electrophoresis. Anal Chem. 1990;62:900–3.

8. Guttman A, Cohen AS, Heiger DN, Karger BL. Analytical and micropreparative ultrahigh resolution of oligonucleotides by polyacrylamide gel high-performance capillary electrophoresis. Anal Chem. 1990;62:137–41.

9. Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, et al. Whole-Genome Random Sequencing and Assembly of Haemophilus influenzae Rd. Science. 1995;269:496–512.

10. Goffeau A, Barrell BG, Bussey H, Davis RW, Dujon B, Feldmann H, et al. Life with 6000 Genes. Science. 1996;274:546–67.

11. Collins FS, Fink L. The Human Genome Project. Alcohol Health Res World. 1995;19:190–5.

12. Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The Sequence of the Human Genome. Science. 2001;291:1304–51.

13. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature. 2001;409:860–921.

14. Metzker ML. Sequencing technologies — the next generation. Nat Rev Genet. 2010;11:31–46.

15. Li R, Fan W, Tian G, Zhu H, He L, Cai J, et al. The sequence and de novo assembly of the giant panda genome. Nature. 2010;463:311–7.

16. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, et al. Real-Time DNA Sequencing from Single Polymerase Molecules. Science. 2009;323:133–8.

17. Mikheyev AS, Tin MMY. A first look at the Oxford Nanopore MinION sequencer. Molecular Ecology Resources. 2014;14:1097–102.

18. Wenger AM, Peluso P, Rowell WJ, Chang P-C, Hall RJ, Concepcion GT, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nat Biotechnol. 2019;37:1155–62.

19. Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, et al. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. Science. 2009;326:289–93.

20. Bickhart DM, Rosen BD, Koren S, Sayre BL, Hastie AR, Chan S, et al. Single-molecule sequencing and chromatin conformation capture enable de novo reference assembly of the domestic goat genome. Nat Genet. 2017;49:643–50.

21. Low WY, Tearle R, Bickhart DM, Rosen BD, Kingan SB, Swale T, et al. Chromosome-level assembly of the water buffalo genome surpasses human and goat genomes in sequence contiguity. Nat Commun. 2019;10:260.

22. Dudchenko O, Batra SS, Omer AD, Nyquist SK, Hoeger M, Durand NC, et al. De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds. Science. 2017;356:92–5.

23. Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, et al. A Whole-Genome Assembly of Drosophila. Science. 2000;287:2196–204.

24. Chevreux B, Wetter T, Suhai S. Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. In: Computer Science and Biology: Proceedings of the German Conference on Bioinformatics(GCB). 1999;99:45–56.

25. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol İ. ABySS: A parallel assembler for short read sequence data. Genome Res. 2009;19:1117–23.

26. Zerbino DR, Birney E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. 2008;18:821–9.

27. Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience. 2012;1:18.

28. Gnerre S, MacCallum I, Przybylski D, Ribeiro FJ, Burton JN, Walker BJ, et al. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proceedings of the National Academy of Sciences. 2011;108:1513–8.

29. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. J Comput Biol. 2012;19:455–77.

30. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res. 2017;gr.215087.116.

31. Kolmogorov M, Yuan J, Lin Y, Pevzner PA. Assembly of long, error-prone reads using repeat graphs. Nat Biotechnol. 2019;37:540–6.

32. Chin C-S, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, et al. Phased diploid genome assembly with single-molecule real-time sequencing. Nat Methods. 2016;13:1050–4.

33. Ruan J, Li H. Fast and accurate long-read assembly with wtdbg2. Nat Methods. 2020;17:155–8.

34. Cheng H, Concepcion GT, Feng X, Zhang H, Li H. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. Nat Methods. 2021;18:170–5.

35. Nurk S, Walenz BP, Rhie A, Vollger MR, Logsdon GA, Grothe R, et al. HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. Genome Res. 2020;30:1291–305.

36. Rautiainen M, Marschall T. MBG: Minimizer-based sparse de Bruijn Graph construction. Bioinformatics. 2021;37:2476–8.

37. Bankevich A, Bzikadze AV, Kolmogorov M, Antipov D, Pevzner PA. Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads. Nat Biotechnol. 2022;40:1075–81.

38. Xu Z, Dixon JR. Genome reconstruction and haplotype phasing using chromosome conformation capture methodologies. Briefings in Functional Genomics. 2020;19:139–50.

39. Boetzer M, Henkel CV, Jansen HJ, Butler D, Pirovano W. Scaffolding pre-assembled contigs using SSPACE. Bioinformatics. 2011;27:578–9.

40. Kajitani R, Toshimoto K, Noguchi H, Toyoda A, Ogura Y, Okuno M, et al. Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. Genome Res. 2014;24:1384-95.

41. Kaplan N, Dekker J. High-throughput genome scaffolding from in-vivo DNA interaction frequency. Nat Biotechnol. 2013;31:1143–7.

42. Marie-Nelly H, Marbouty M, Cournac A, Flot J-F, Liti G, Parodi DP, et al. High-quality genome (re)assembly using chromosomal contact data. Nat Commun. 2014;5:5695.

43. Baudry L, Guiglielmoni N, Marie-Nelly H, Cormier A, Marbouty M, Avia K, et al. instaGRAAL: chromosome-level quality scaffolding of genomes using a proximity ligation-based scaffolder. Genome Biology. 2020;21:148.

44. Burton JN, Adey A, Patwardhan RP, Qiu R, Kitzman JO, Shendure J. Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. Nat Biotechnol. 2013;31:1119–25.

45. Ghurye J, Pop M, Koren S, Bickhart D, Chin C-S. Scaffolding of long read assemblies using long range contact information. BMC Genomics. 2017;18:527.

46. Ghurye J, Rhie A, Walenz BP, Schmitt A, Selvaraj S, Pop M, et al. Integrating Hi-C links with assembly graphs for chromosome-scale assembly. PLOS Computational Biology. 2019;15:e1007273.

47. Brinton J, Ramirez-Gonzalez RH, Simmonds J, Wingen L, Orford S, Griffiths S, et al. A haplotype-led approach to increase the precision of wheat breeding. Commun Biol. 2020;3:1–11.

48. Tewhey R, Bansal V, Torkamani A, Topol EJ, Schork NJ. The importance of phase information for human genomics. Nat Rev Genet. 2011;12:215–23.

49. Glusman G, Cox HC, Roach JC. Whole-genome haplotyping approaches and genomic medicine. Genome Medicine. 2014;6:73.

50. Garg S. Computational methods for chromosome-scale haplotype reconstruction. Genome Biology. 2021;22:101.

51. Koren S, Rhie A, Walenz BP, Dilthey AT, Bickhart DM, Kingan SB, et al. De novo assembly of haplotype-resolved genomes with trio binning. Nat Biotechnol. 2018;36:1174-82.

52. Bansal V, Bafna V. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. Bioinformatics. 2008;24:i153–9.

53. Edge P, Bafna V, Bansal V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome Res. 2017;27:801–12.

54. Selvaraj S, R Dixon J, Bansal V, Ren B. Whole-genome haplotype reconstruction using proximity-ligation and shotgun sequencing. Nat Biotechnol. 2013;31:1111–8.

55. Kajitani R, Yoshimura D, Okuno M, Minakuchi Y, Kagoshima H, Fujiyama A, et al. Platanus-allee is a de novo haplotype assembler enabling a comprehensive access to divergent heterozygous regions. Nat Commun. 2019;10:1702.

56. Kronenberg ZN, Rhie A, Koren S, Concepcion GT, Peluso P, Munson KM, et al. Extended haplotype-phasing of long-read de novo genome assemblies using Hi-C. Nat Commun. 2021;12:1935.

57. Zhang X, Zhang S, Zhao Q, Ming R, Tang H. Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data. Nat Plants. 2019;5:833–45.

58. Cheng H, Jarvis ED, Fedrigo O, Koepfli K-P, Urban L, Gemmell NJ, et al. Haplotype-resolved assembly of diploid genomes without parental data. Nat Biotechnol. 2022;40:1332–5.

59. Garg S, Fungtammasan A, Carroll A, Chou M, Schmitt A, Zhou X, et al. Chromosome-scale, haplotype-resolved assembly of human genomes. Nat Biotechnol. 2021;39:309–12.

60. Ouchi S, Kajitani R, Itoh T. GreenHill: a de novo chromosome-level scaffolding and phasing tool using Hi-C. Genome Biology. 2023;24:162.

61. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34:3094–100.

62. Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979;9:62–6.

63. Rhie A, McCarthy SA, Fedrigo O, Damas J, Formenti G, Koren S, et al. Towards complete and error-free genome assemblies of all vertebrate species. Nature. 2021;592:737–46.

64. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics. 2011;27:764–70.

65. Ranallo-Benavidez TR, Jaron KS, Schatz MC. GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes. Nat Commun. 2020;11:1432.

66. Roach MJ, Schmidt SA, Borneman AR. Purge Haplotigs: allelic contig reassignment for third-gen diploid genome assemblies. BMC Bioinformatics. 2018;19:460.

67. Guan D, McCarthy SA, Wood J, Howe K, Wang Y, Durbin R. Identifying and removing haplotypic duplication in primary genome assemblies. Bioinformatics. 2020;36:2896–8.

68. Durand NC, Shamim MS, Machol I, Rao SSP, Huntley MH, Lander ES, et al. Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. cels. 2016;3:95–8.

69. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv: 2013. doi:arXiv:1303.3997v2[q-bio.GN].

70. Kadota M, Nishimura O, Miura H, Tanaka K, Hiratani I, Kuraku S. Multifaceted Hi-C benchmarking: what makes a difference in chromosome-scale genome scaffolding? GigaScience. 2020;9:giz158.

71. mapping_pipeline. https://github.com/ArimaGenomics/mapping_pipeline. Accessed 26 Apr 2022.

72. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25:2078–9.

73. Picard. https://broadinstitute.github.io/picard. Accessed 26 Apr 2022.

74. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2.

75. Rhie A, Walenz BP, Koren S, Phillippy AM. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. Genome Biology. 2020;21:245.

76. Ewing B, Hillier L, Wendl MC, Green P. Base-Calling of Automated Sequencer Traces UsingPhred. I. Accuracy Assessment. Genome Res. 1998;8:175–85.

77. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics. 2012;28:593–4.

78. DeMaere MZ, Darling AE. Sim3C: simulation of Hi-C and Meta3C proximity ligation sequencing technologies. GigaScience. 2018;7:gix103.

79. Marçais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, Zimin A. MUMmer4: A fast and versatile genome alignment system. PLOS Computational Biology. 2018;14:e1005944.

80. Renschler G, Richard G, Valsecchi CIK, Toscano S, Arrigoni L, Ramírez F, et al. Hi-C guided assemblies reveal conserved regulatory topologies on X and autosomes despite extensive genome shuffling. Genes Dev. 2019;33:1591–612.

81. Robinson GE, Hackett KJ, Purcell-Miramontes M, Brown SJ, Evans JD, Goldsmith MR, et al. Creating a Buzz About Insect Genomes. Science. 2011;331:1386–1386.

82. Zhang G. Bird sequencing project takes off. Nature. 2015;522:34–34.

83. Lewin HA, Richards S, Lieberman Aiden E, Allende ML, Archibald JM, Bálint M, et al. The Earth BioGenome Project 2020: Starting the clock. Proceedings of the National Academy of Sciences. 2022;119:e2115635118.

84. Durand NC, Robinson JT, Shamim MS, Machol I, Mesirov JP, Lander ES, et al. Juicebox Provides a Visualization System for Hi-C Contact Maps with Unlimited Zoom. Cell Syst. 2016;3:99–101.

85. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. Nat Biotechnol. 2018;36:338–45.

86. Jain M, Olsen HE, Turner DJ, Stoddart D, Bulazel KV, Paten B, et al. Linear assembly of a human centromere on the Y chromosome. Nat Biotechnol. 2018;36:321–3.

87. Miga KH, Koren S, Rhie A, Vollger MR, Gershman A, Bzikadze A, et al. Telomere-to-telomere assembly of a complete human X chromosome. Nature. 2020;585:79–84.

88. Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze AV, Mikheenko A, et al. The complete sequence of a human genome. Science. 2022;376:44–53.

謝辞

本研究は東京工業大学 伊藤武彦教授のご指導のもとで行われました。伊藤教授には心より感謝申し上げます。

GreenHill のソースコードの改良やベンチマークの実施にご協力頂きました東京工業大学 梶谷嶺助教(研究当時)に感謝申し上げます。

多くの事務手続きでお世話になりました伊藤研究室秘書 坂東由衣さんに感謝申し上げます。研究の過程で有意義な議論をしてくださった伊藤研究室の皆様に感謝申し上げます。