bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Describing the Local Structure of Sequence Graphs

Yohei Rosen[1,2], Jordan Eizenga[2], and Benedict Paten[2]

[1] New York University School of Medicine, 550 1st Avenue, New York, NY 10016, United States yohei.rosen@nyumc.org
[2] University of California Santa Cruz Genomics Institute, 1156 High Street, Mailstop CBSE, Santa Cruz, CA 95064, United States benedict@soe.ucsc.edu

Abstract. Analysis of genetic variation using graph structures is an emerging paradigm of genomics. However, defining genetic sites on sequence graphs remains an open problem. Paten's invention of the *ultrabubble* and *snarl*, special subgraphs of sequence graphs which can identified with efficient algorithms, represents important first step to segregating graphs into genetic sites. We extend the theory of ultrabubbles to a special subclass where every detail of the ultrabubble can be described in a series and parallel arrangement of genetic sites. We furthermore introduce the concept of *bundle* structures, which allows us to recognize the graph motifs created by additional combinations of variation in the graph, including but not limited to runs of abutting single nucleotide variants. We demonstrate linear-time identification of bundles in a bidirected graph. These two advances build on initial work on ultrabubbles in bidirected graphs, and define a more granular concept of genetic site.

Keywords: Sequence graphs, genetic variants

1 Background

The concept of the genetic site underpins both classical genetics and modern genomics. From a biological perspective, a site is a position at which mutations have occurred in different samples' histories, leading to genetic variation. From an engineering perspective, a site is a subgraph with left and right endpoints where traversals by paths correspond to alleles. This is useful for indexing and querying variants in paths and for describing variants in a consistent and granular manner.

Against a linear reference, it is trivial to define sites, provided that we disallow variants spanning overlapping positions. This is clearly demonstrated by VCF structure [1]. VCF sites, consisting of any number of possible alleles, are identified by their endpoints with respect to the linear reference.

If we wish to analyze a set of variants containing structural variation, highly divergent sequences or nonlinear references structures, then a linear reference with only non-overlapping variants is no longer a sufficient model. Datasets with

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

one or more of these properties are becoming more common [2, 3], and sequence graphs [4] have been developed as a method of representing them. However, defining sites on graphs is considerably more difficult than on linear reference structures and the creation of methods to fully decompose sequence graphs into sites remains an unsolved problem.

2 The Challenges of Defining Sites on Graphs

On a graph-based reference, the linear reference definition of a site as a position along the reference and a set of alleles fails to work for several reasons:

1. Sequences which are at the same location in linear position may not have comparable contexts. This is a consequence of having variants which cannot be represented as edits to the linear reference but rather as edits to another variant. We illustrate this with an example from *1000 Genomes* polymorphism data, visualized using Sequence Tube Maps [5].

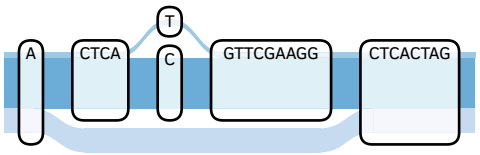

Fig. 1. The context of the single nucleotide variant shown does not exist in all variants spanning its linear position

2. Elements of sequence may not be linearly ordered. Parallel structure of the graph (3.) is one sort of non-linearity. Graphs also allow repetitive, inverted or transposed elements of sequence. These all prevent linear ordering.

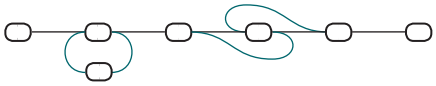

Fig. 2. A cycle and an inversion in a graph

3. The positions spanned by different elements of variation may partially overlap. Therefore, multiple mutually exclusive segments of sequence in a region

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

of the graph cannot be considered to be alternates to each other at a well-defined position without having to include extraneous sequence that is shared between some but not all of the "alleles."

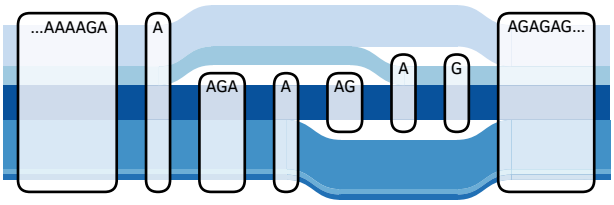

Fig. 3. Overlapping deletions, from 1000 Genomes polymorphism data

We can expect that the density of these graph structures will increase with increasing population sizes included in datasets.

Our aim will be to recognize and fully decompose subgraphs resembling Example 1 into a notion of site, and isolate these from elements of the graph resembling Examples 2 and 3.

3 Mathematical Background

3.1 Directed and Bidirected Sequence Graphs

The graphs used to represent genetic information consist of labelled nodes and edges. Nodes are labelled with sequence fragments. Edges form paths whose labels spell out allowed sequences. Two types of graph are used.

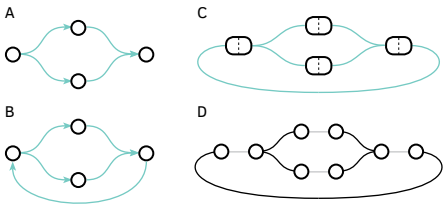

Fig. 4. (A) A directed acyclic graph (B) A cyclic directed graph (C) Graph B represented as a bidirected graph. This cycle is proper. (D) Graph C represented as a biedged graph

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

The more simple type is the directed graph. A directed graph (or "digraph") G consists of a set V of nodes and a set E of directed edges. A directed edge is an ordered tuple (x, y), consisting of a *head* $x \in V$ and *tail* $y \in V$. A directed path is a sequence of nodes joined by edges, followed head to tail. G is a *directed acyclic graph* (DAG) if it admits no directed path which revisits any node.

A bidirected graph G [6] consists of a set V of vertices and a set E of edges. Each vertex $v \in V$ consists of a pair of *node-sides* $\{v_{left}, v_{right}\}$ and each edge is an unordered tuple of node-sides. Bidirected graphs have the advantage of being able to represent inversion events.

We write N for the set of node-sides in the bidirected graph G. The *opposite* $\hat{n}$ of a node-side n is the other node-side at the same vertex as n.

A sequence $p = x_1, x_2, \ldots, x_k$ of node-sides is a *path* if $\forall x_i$,

1. if $x_{i-1} \neq \hat{x_i}$, then $x_{i+1} = \hat{x_i}$
2. if $x_{i-2} = \hat{x_{i-1}}$, then $\{x_{i-1}, x_i\} \in E$
3. any contiguous subsequence of p consisting of a node-side x alternating with its opposite $\hat{x}$ must either be even-numbered in length or must be a prefix or suffix of p

Informally, this means that in a path, consecutive pairs forming edges alternate with pairs of opposite node-sides or, equivalently, that paths visit both node-sides of the vertices they pass through. They can however begin or on an isolated node-side.

A bidirected graph G is *cyclic* if it admits a path visiting a node-side twice. Therefore the self-incident hairpin motif (below, right) is considered a cycle. A bidirected graph G is *properly cyclic* if it admits a path which visits a pair $\{n, \hat{n}\}$ twice in the same order.

Fig. 5. (Left) A properly cyclic graph. (Right) The self-incident hairpin motif of a cyclic but not properly cyclic graph

Some publications refer to biedged graphs. These are $\{black, grey\}$-edge-colored undirected graphs, where every node is paired with precisely one other by sharing a grey edge and paths in the graph must alternate between traversing black and grey edges. Paten elaborates on this construction in [7] and shows that it is equivalent to a bidirected graph. We will restrict our language to that of bidirected graphs, recognizing that these are equivalent to biedged graphs.

Acyclic bidirected graphs are structurally equivalent to directed graphs in that

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Lemma 1. *If G is a bidirected acyclic graph, there exists an isomorphic directed acyclic graph $D(G)$.*

Proof. See [7]

3.2 Bubbles, Superbubbles, Ultrabubbles and Snarls

The first use of local graph structure to identify variation was the detection of *bubbles* [8] in order to detect and remove sequencing errors from assembly graphs. Their bubble is the graph motif consisting of two paths which share a source and a sink but are disjoint between.

The general concept of bubbles was extended by Onodera et al, who defined superbubbles in directed graphs [9]. Brankovic demonstrates an $\mathcal{O}(|V| + |E|)$ algorithm to identify them [10], building off work of Sung [11].

We restate the Onodera definition, modified slightly as to be subgraph-centric rather than boundary-centric: A subgraph $S \subseteq G$ of a directed graph is a *superbubble with boundaries* (s, t) if

1. (reachability) t is reachable from s by a directed path in S
2. (matching) the set of vertices reachable from s without passing through t is equal to the set of vertices from which t is reachable without passing through s, and both are equal to S
3. (acyclicity) S is acyclic
4. (minimality) there exists no $t' \in S$ such that boundaries (s, t') fulfil 1,2 and 3. There exists no $s' \in S$ such that (s', t) fulfil 1,2 and 3.

To motivate our definition of a superbubble equivalent on bidirected graphs, we prove some consequences of the matching property.

Proposition 2. *Let $S \subseteq G$ be a subgraph of a directed graph. If S possesses the matching property relative to a pair (s, t), then it possesses the following three properties:*

1. *(2-node separability) Deletion of all incoming edges of s and all outgoing edges of t disconnects S from the remainder of the graph.*
2. *(tiplessness) There exist no node $n \in S \backslash \{s, t\}$ such that n has either only incoming or outgoing edges.*
3. *S is weakly connected*

Proof. (*matching $\Rightarrow$ separability*) *Suppose $\exists x \notin S, y \in S \backslash \{s, t\}$ such that there exists either an edge $x \to y$ or an edge $y \to x$. Suppose wlog that $\exists$ an edge $x \to y$. By matching, there exists a path $y \to \cdots \to t$ without passing through s. We can then construct the path $x \to y \to \cdots \to t$ which does not pass through s. But by matching this implies that $x \in S$, which leads to a contradiction.*

The converse need not be true on directed graphs[3]. We define two structures on bidirected graphs. The first is the ultrabubble, which given Proposition 2,

[3] It is on bidirected graphs

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

can be thought of as an analogue to a superbubble. The second, the snarl, is a more general object which preserves the property of 2-node separability from the larger graph without having strong guarantees on its internal structure. The following definitions are due to Paten [7]:

A connected subgraph $S \subseteq G$ of a bidirected graph G is a *snarl* (S, s, t) with boundaries (s, t), if

1. $s \neq \hat{t}$
2. (2-node separability) every path between a pair of node-sides in $x \in S, y \in G \backslash S$ contains either $s \to \hat{s}$ or $t \to \hat{t}$ as a subpath.
3. (minimality) there exists no $t' \in S$ such that boundaries (s, t') fulfil 1 and 2. There exists no $s' \in S$ such that (s', t) fulfil 1 and 2

The class of *ultrabubbles* is the subclass of snarls (S, s, t) furthermore fulfilling

4. S is acyclic
5. S contains no tips — vertices having one node-side involved in no edges

Three examples of ultrabubbles are shown below.

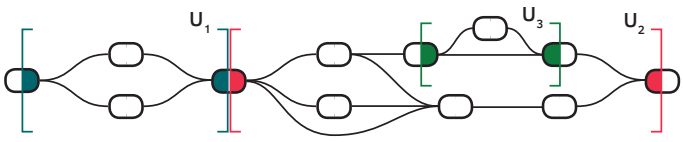

Fig. 6. Three ultrabubbles, boundaries colored blue, pink and green. These illustrate the non-overlapping property

The following is important property of snarls.

Proposition 3 (Non-overlapping property). *If two distinct snarls share a vertex (node-side pair) then either they share a boundary node or one snarl is included in the other's interior.*

Proof. Let S be a snarl with boundaries s, t. Let T be another snarl, with boundaries u, v. Suppose that $u \in S \backslash \{s, t\}$ but $v \notin S$, and $s \notin T$.

Consider the set $S \cap T$. It is nonempty since it contains u. Let $x \in S \cap T$. Let $y \notin S \cap T$. Suppose that there exists a path $p = x \leftrightarrow \cdots \leftrightarrow y$ which neither passes through u nor t.

Since $y \notin S \cap T$, either $y \notin S$ or $y \notin T$. Wlog, assume $y \notin T$. Then due to the separability of T, since the path p does not pass through u, it must pass through v before leaving T to visit y. But $v \notin S$ so p must also pass through s before leaving S to visit v since it does not pass through t. But it must pass through v before leaving T to visit s, which leads to an impossible sequence of events. Therefore any path $x \leftrightarrow \cdots \leftrightarrow y$ for $x \in S \cap T, y \notin S \cap T$ must pass through either u or t. This contradicts the minimality of both S and T.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

This non-overlapping property is also a nesting property. Observe that, due to Proposition 3, the relation $U \leq V$ on snarls U, V defined such that $U \leq V$ if U is entirely contained in V has the property that if $U \leq V$ and $U \leq W$, then either $V \leq W$ or $W \leq V$. Therefore the partial order on the snarls of G defined by the relation $\leq$ will always be equivalent to a tree diagram. A *bottom level* snarl is one which forms a leaf node of this tree.

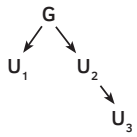

Fig. 7. The nesting tree diagram of the ultrabubbles from the previous figure. U_1 and U_3 are bottom-level

The equivalent of Proposition 3 for superbubbles was stated without proof by Onodera in [9]. Our proof also constitutes a proof of the statement for superbubbles, due to the following proposition, proven by Paten in [7]:

Proposition 4. *Every superbubble in a directed graph corresponds to an ultrabubble in the equivalent (see Lemma 1) bidirected graph.*

Identifying all superbubbles in a directed graph or all snarls in a bidirected graph introduces a method of compartmentalizing a graph into partitions whose contents are all in some sense at the same position in the graph, and for which the possible internal paths are independent of what path they continue on beyond their boundaries. We will use this concept to define sites for certain specialized classes of graphs.

4 Graphs which are Decomposable into Nested Simple Sites

We will extend the theory of ultrabubbles to a theory of nested sites where the structure of certain graphs can be fully described in terms of combinations of linear orderign and ultrabubble nesting relationships. This is important for

1. Identifying nested variation
2. Indexing traversals

4.1 Traversals and Subpaths

An (s, t)-*traversal* of S is a path in S beginning with s and ending with t. An (s, s)-*traversal* and a (t, t)-*traversal* are analogously defined. Presence of an

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

(s, s)- or (t, t)-traversal implies cyclicity. Two traversals of a snarl are *disjoint* if they are disjoint on $S \backslash \{s, t\}$.

Paten's [7] snarls and ultrabubbles are 2-node separable subgraphs whose paired boundary nodes isolate their traversals from the larger graph. We can state this with more mathematical rigor:

Claim. Consider a snarl (S, s, t) in a bidirected graph G. The set of all paths in G which contain a single (s, t)-traversal as contiguous a subpath is isomorphic to the set-theoretic product $P(s) \times Trav(s, t) \times P(t)$ consisting of the three sets

1. $P(s) := \{\text{paths in } G \backslash S \text{ terminating in } \hat{s}\}$
2. $Trav(s, t) := \{(s, t)\text{-traversals of } S\}$
3. $P(t) := \{\text{paths in } G \backslash S \text{ beginning with } \hat{t}\}$

The isomorphism is the function mapping $p_1 \in P(s), p_2 \in Trav(s, t), p_3 \in P(t)$ to their concatenation $p_1 p_2 p_3$.

This property is important because it allows us to express the set of all haplotypes traversing a given linear sequence of snarls in terms of combinations of alleles for which we do not need to check if certain combinations are valid.

4.2 Simple Bubbles and Nested Simple Bubbles

Definition 5. *An ultrabubble (S, s, t) is a simple bubble if all (s, t)-traversals are disjoint.*

Simple bubbles are structurally equivalent to (multiallelic) sites consisting of disjoint substitutions, insertions or deletions, with all alleles spanning the same boundaries.

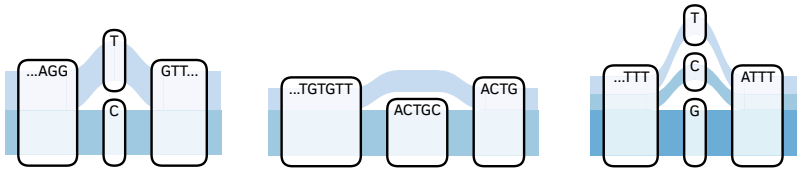

Fig. 8. Three examples of simple bubbles from the 1000 Genomes graph

Proposition 7 below demonstrates that we can identify simple bubbles in $\mathcal{O}(|V|)$ time given that we have found all snarl boundaries. Paten has shown [7] that identification of snarl boundaries is achieved in $\mathcal{O}(|E| + |V|)$ time. To find the ultrabubbles among these, note that checking for acyclicity is $\mathcal{O}(|E| + |V|)$ on account of the unbranching nature of these snarls' interiors.

Given a node-side n, write $Nb(n)$ for the set of all neighbors of n. Note that $a \in Nb(b) \Leftrightarrow b \in Nb(a)$.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Lemma 6 (Nodes in an ultrabubble are orientable with respect to the ultrabubble boundaries). *Given an ultrabubble (S, s, t) and given $n \in S \backslash \{s, t\}$, consider the set T of all (s, t)-traversals of S passing through n. Then either*

1. *$\forall p \in T$, an element of $Nb(n)$ precedes n in p*
2. *$\forall p \in T$, an element of $Nb(n)$ follows n in p*

In the former case we call n s-sided, otherwise we call it t-sided.

Proof. *This is a corollary to Lemma 1.*

Proposition 7 (Simple bubbles have unbranching interiors). *Let (S, s, t) be an ultrabubble. Then all traversals are disjoint iff every interior node-side has precisely one neighbor.*

Proof. *($\Rightarrow$) Suppose that all (s, t)-traversals of S are disjoint. Suppose $\exists$ a node-side $n \in S \backslash \{s, t\}$ with multiple neighbors.*

Since n is orientable with respect to (s, t), suppose, without loss of generality, that it is s-sided. Then there exist distinct paths from s to n passing through each of its neighbors. Continuing these with a path from n^{opp} to t produces two nondisjoint traversals of S.

($\Leftarrow$) Suppose that every interior node-side has precisely one neighbor. Suppose that there exist two distinct nondisjoint traversals of S. For no node-side to have multiple neighbors, they must coincide at every node-side, contradicting the assumption that they are not the same traversal.

We seek to extend this simple property to more complex graph structures. We will take advantage of the nesting of nondisjoint ultrabubbles proven in Proposition 3 to define another structure in which nondisjoint traversals are easily indexed.

Definition 8. *An ultrabubble $(S, s, t) \subseteq G$ is decomposable into nested simple sites if either:*

1. *S is a simple bubble*
2. *if, for every ultrabubble S' contained in the interior of S, you replace the ultrabubble with a single edge $s - t$ whenever S' is decomposable into simple sites, then S becomes a simple bubble*

The following figure demonstrates decomposability into nested simple sites.

Proposition 9. *If an ultrabubble (U, s, t) is decomposable into nested simple sites, then the complete node sequence of any (s, t)-traversal can be determined only by specifying the path it takes inside those nested ultrabubbles within which the traversal does not visit any further nested ultrabubble.*

Proof. *Let p be a (s, t)-traversal of an ultrabubble U which is decomposable into nested simple sites. Let V be a nested ultrabubble inside U. If p traverses, V, write $p|_V$ for the traversal p restricted to V.*

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

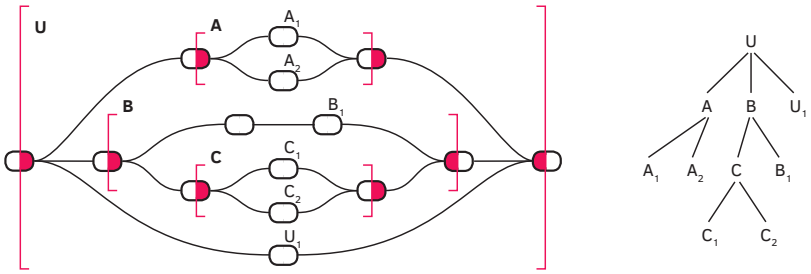

Fig. 9. Left: A nesting of four ultrabubbles. Right: The tree structure to index traversals of U implied by Proposition 9

Suppose that $t|_V$ intersects no nested ultrabubbles within V. Then $t|_V$ is disjoint of all other traversals within V due to U begin decomposable into nested simple sites. Therefore specifying any node of $t|_V$ uniquely identifies it.

Suppose that $t|_V$ intersects some set of ultrabubbles nested within V. Since U is decomposable into nested simple sites, the nodes of $t|_V$ must be linear and disjoint of all other paths if we replace all ultrabubbles nested in V with edges joining their boundaries. Therefore specifying which ultrabubbles are crossed uniquely determines the nodes included in $t|_V$ which lie outside of the nested ultrabubbles in V.

The statement of the proposition follows from the two arguments above by induction.

Proposition 10. *An ultrabubble is decomposable into nested simple sites iff every node side is either the interior ultrabubble boundary or has precisely one neighbor.*

Proof. This can be established using Proposition 7.

This property allows $\mathcal{O}(|V| + |E|)$ evaluation of whether a graph is decomposable into nested simple sites, by arguments analogous to those for simple bubbles.

4.3 A Partial Taxonomy of Graph Notifs which do not Admit Decomposition into Sites

In section 4.3, we will show that we can decompose a graph into nested simple sites as defined in the previous section if it lacks a certain forbidden motif. We will begin with examples of three graph motifs, and the biological events which might produce them.

We describe some graph features which prevent decomposition into nested sites below, and the sets of mutations which might have produced them.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

1. Two (or more) substitutions or deletions against a linear sequence which overlap, but not completely.

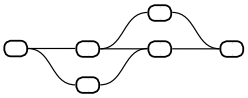

Fig. 10. Overlapping substitutions (or deletions)

2. A substitution (or deletion) which spans elements of sequence on the interior of two disjoint ultrabubbles. Addition of such an edge joining two ultrabubbles which were decomposable into nested simple sites will consolidate the two into a single ultrabubble which is not decomposable into nested simple sites.

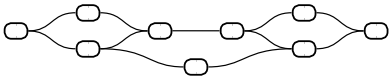

Fig. 11. An edge crossing bubble boundaries.

3. Two SNVs or other simple elements of variation at adjacent positions. This will be the focus of our Section 5.

4.4 The Relationship Between Nested Simple Sites and Series Parallel Graphs

The structure of ultrabubbles decomposable into nested simple sites, and their tree representation (see Fig 7) might be familiar to the graph theorist familiar with series-parallel digraphs. The fact that the digraphs equivalent to ultrabubbles form a subclass of the two-terminal series-parallel digraphs is interesting due to the computational properties of the latter class of graphs.

Definition 11. *A directed graph G is two-terminal series parallel (TTSP) with source s and sink t if either*

1. *G is the two-element graph with a single directed edge $s \to t$*
2. *There exist TTSP graphs G_1, G_2 with sources s_1, s_2 and sinks t_1, t_2 such that G is formed from G_1, G_2 by identification of s_1 with s_2 as s and identification of t_1 with t_2 as t (Parallel addition)*

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

3. *There exist TTSP graphs G_1, G_2 with sources s_1, s_2 and sinks t_1, t_2 such that G is formed from G_1, G_2 by identification of t_1 with s_2 (Series addition)*

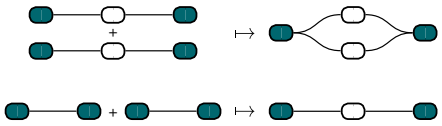

Fig. 12. Top: parallel addition. Bottom: series addition

Two terminal series parallel digraphs have a useful forbidden subgraph characterization.

Proposition 12 (From [12]). *A directed graph G is two terminal series parallel if and only if it contains no subgraph homeomorphic to the graph W shown below Proof: Refer to Valdes [12] and Duffin [13]*

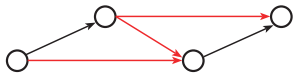

Fig. 13. The W motif

Proposition 13. *If an ultrabubble (U, s, t) is decomposable into nested simple sites, then the equivalent directed graph is TTSP with source s and sink t.*

Proof. Suppose that the directed graph $D(U)$ equivalent to U (which exists by Lemma 1) contains a subgraph homeomorphic to W. Then there must be a node-side u in U with two neighbours a_1, a_2 which are the beginnings of disjoint paths p_1, p_2 ending on node-sides b_1, b_2 which are neighbours of a node-side v. By Proposition 10, u and v must be ultrabubble boundaries. Since p_1, p_2 are disjoint, u and v must be opposing boundaries of the same ultrabubble. But the presence of a subgraph homeomorphic to W also implies that there exists a pair q_1, q_2 of disjoint paths, one from a node x to $\hat{u}$ and the other from x to v, both not passing through u or $\hat{v}$. But this is not possible since it would contradict 2-node separability of (u, v).

We highlight the middle "Z-arm" of the W-motif in our first two examples of ultrabubbles which are not decomposable into nested simple sites.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

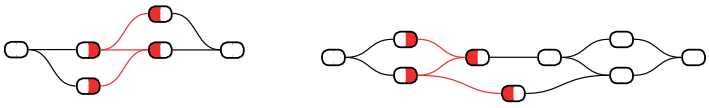

Fig. 14. Portions of the ultrabubbles 1. and 2. of section 4.2, showing the nodes which project to the forbidden subgraph W

5 Abutting Variants

We wish to decompose the graph structure of sets of variants lying at adjacent positions such that there is no conserved sequence between them able to form an ultrabubble boundary. We will define a graph motif called the *balanced recombination bundle* which corresponds this graph structure, and can be rapidly detected.

We observe examples abutting single nucleotide variants (SNVs) in the 1000 Genomes polymorphism data. It is a reasonable hypothesis that these should become more common as the population sizes of sequencing datasets increases, since, statistically, the distribution of variation across the genome should grow less sparse as the population increases.

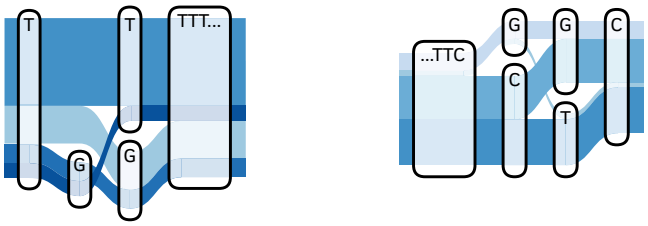

Fig. 15. Two examples of abutting SNVs in the 1000 Genomes graph

5.1 Bundles

Definition 14. *An internal chain* $n_1 \to n_2 \to \cdots \to n_k$ *is a sequence of node-sides such that* $\forall i, 2 \leq i \leq k, n_i \in Nb(n_{i-1})$.

Definition 15. *We say that a tuple* (L, R) *of sets of node-sides is a bundle if*

1. *(Matching)* $\forall \ell \in L, Nb(\ell) \subseteq R$ *and* $Nb(\ell) \neq \varnothing$*;* $\forall r \in R, Nb(r) \subseteq L$ *and* $Nb(r) \neq \varnothing$

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

2. *(Connectedness)* $\forall \ell \in L, r \in R$, *there exists an internal chain* $\ell \to r_1 \to \ell_1 \to \cdots \to r_k \to \ell_k \to r$ *such that* $\forall i, 1 \leq i \leq k, r_i \in R$ *and* $\ell_i \in L$

Definition 16. *We say that a tuple* (L, R) *of sets of node-sides is a balanced recombination bundle (R-bundle for short) if*

1. *(Complete matching)* $\forall \ell \in L, Nb(\ell) = R$ *and* $\forall r \in R, Nb(r) = L$
2. *(Acyclicity)* $L \cap R = \varnothing$

Lemma 17. *A balanced recombination bundle is a bundle.*

Proof. Complete matching $\Rightarrow$ *matching.*

Complete matching $\Rightarrow$ *connectedness by the chain* $\ell \to r$ *for all* $\ell \in L, r \in R$

Definition 18. *An unbalanced bundle is a bundle which is not a balanced recombination bundle. An unbalanced bundle is acyclic if* $L \cap R = \varnothing$.

Definition 19. *We say that two bundles* $(L_1, R_1), (L_2, R_2)$ *are isomorphic if either* $L_1 = L_2$ *and* $R_1 = R_2$ *or* $L_1 = R_2$ *and* $R_1 = L_2$.

We will describe a $\mathcal{O}(|V| + |E|)$ algorithm to detect and categorize bundles exhaustively for all node-sides in a bidirected graph. To establish the validity of this algorithm, we need several preliminary results:

Lemma 20. *Every* $q \in N$ *is either a tip or an element of a bundle.*

Proof. Suppose that q *is not a tip. Define a function* W *that maps a tuple* (L, R) *of nonempty sets of node-sides to a tuple* $W(L), W(R)$ *where*

$$W(R) := R \cup \bigcup_{\ell \in L} Nb(\ell)$$
$$W(L) := L \cup \bigcup_{r \in W(R)} Nb(r)$$

$\forall n \in \mathbb{N}$ *define*

$$W^n((L, R)) := \underbrace{W \circ \cdots \circ W((L, R))}_{n \text{ times}}$$
$$W^\infty((L, R)) := W^k((L, R)) \text{ for } k \text{ such that } W^{k+i}((L, R)) = W^k((L, R)) \, \forall i \in \mathbb{N}$$

W^∞ *exists since* W^n *is nondecreasing with respect to set inclusion and our graphs are finite. Now define* $\overline{W}(q) := W^\infty((\{q\}, Nb(q)))$, *noting that* $Nb(q) \neq \varnothing$ *since* $\{q\}$ *is not a tip. Let us write* L_{W^∞} *and* R_{W^∞} *for the respective elements of* $\overline{W}(q)$. *We claim that* $\overline{W}(q)$ *is a bundle.*

Proof of matching: let $\ell \in L_{W^\infty}, r \in R_{W^\infty}$. *By construction of* W,

$$Nb(\ell) \subseteq W(R_{W^\infty}) = R_{W^\infty}$$
$$Nb(r) \subseteq W(L_{W^\infty}) = L_{W^\infty}$$

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Proof of connectedness: let $\ell \in L_{W^\infty}, r \in R_{W^\infty}$. We will show that for any $r \in R_{W^\infty}$, $\exists$ an internal chain $q \to r_1 \to \ell_1 \to \cdots \to r_k \to \ell_k \to r$ such that $\forall i, 1 \leq i \leq k, r_i \in R_{W^\infty}$ and $\ell_i \in L_{W^\infty}$.

Suppose that $r \in Nb(q)$, then we are done. Otherwise, since $r \in R_{W^\infty}$, there exists some minimal $n \in \mathbb{N}$ such that $r \in$ the R-set R_{W^n} of some $W^n((\{q\}, Nb(q)))$. It is straightforward to see that we can then construct an internal chain $q \to r_0 \to \ell_1 \to r_1 \to \ldots \ell_{n-1} \to r$ such that $\forall i, 1 \leq i \leq n-1, r_i \in R_{W^i}, \ell_i \in L_{W^i}$. By an analogous argument, we can do the same for an internal chain $\ell \to \cdots \to r'$ for some $r' \in Nb(q)$. Concatenation of the first chain with the reverse of the second gives our chain $\ell \to \cdots \to r$, proving connectedness.

Proposition 21. *If $q \in L$ for a bundle (L, R), then $(L, R) = \overline{W}(q)$*

Proof. Suppose that $\overline{W}(q) \neq (L, R)$. Then either $L \neq L_{W^\infty}$ or $R \neq R_{W^\infty}$. First, suppose the latter. Suppose that $\exists r \in R$ such that $r \notin R_{W^\infty}$. Since (L, R) is a bundle, we know that there is an internal chain $q \to r_0 \to \ell_1 \to r_1 \to \cdots \to r_k \to \ell_k \to r$ with all $r_i \in R$, $\ell_i \in L$. But, using the same shorthand as before, it is also evident that $r_i \in R_{W^i}, \ell_i \in L_{W^i}\, \forall i, 1 \leq i \leq k$. But since $\ell_k \in Nb(r)$, we can deduce that $r \in R_{W^{k+1}}$, which leads to a contradiction since $r \notin R_{W^\infty}$.

Suppose otherwise that $\exists r \in R_{W^\infty}$ such that $r \notin R$. Consider an internal chain $c = q \to r_0 \to \ell_1 \to r_1 \to \cdots \to r_k \to \ell_k \to r$ fulfilling the conditions needed to prove connectedness of $\overline{W}(q)$. Note that $q \in L$ and by matching $r_0 \in Nb(q)$. But $r \notin R$, which leads to a contradiction since it means that there must exist two consecutive members somewhere in the chain c which cannot be neighbors.

We say that a node-side n is *involved in* a bundle (L, R) if $n \in L$ or $n \in R$.

Corollary 22 (To Proposition 21). *Every non-tip node-side is involved in precisely one bundle.*

5.2 An Algorithm for Bundle-Finding

The diagram in Fig 16 demonstrates our algorithm for finding the balanced recombination bundle containing a query node-side q if it is contained in one, and discovering that it is not if it is not. The is written in pseudocode below, with an illustration following.

In order to prove that this is a valid algorithm for detection of balanced recombination bundles, we need the following lemma.

Lemma 23. *Let (L, R) be a tuple of sets of node-sides. If $\exists q \in L$ such that $\forall a \in Nb(q), \forall b \in Nb(a), Nb(b) \subseteq Nb(q)$ but $Nb(q) \subset R$, then (L, R) cannot be connected (in the sense of Definition 15).*

Proof. Let $B = \bigcup_{a \in Nb(q)} Nb(a)$. We know that $\forall b \in B, Nb(b) \subseteq Nb(q)$. Suppose that (L, R) is connected. Choose $r \in R \backslash Nb(q)$. Then $\exists$ an internal chain $c = q \to r_1 \to \ell_1 \to \cdots \to r_k \to \ell_k \to r$ with $r_i \in R, \ell_i \in L\, \forall i$. Since $q \in B$, $Nb(b) \subseteq Nb(q)\, \forall b \in B$, and $Nb(a) \subseteq B\, \forall a \in Nb(q)$, it is impossible that the

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Algorithm 1: Balanced recombination bundle finding

Data: Node-side q
Result: Bundle containing q if it is in a balanced recombination bundle, $\varnothing$ if q is in an unbalanced bundle or is a tip
begin
  if $Nb(q) = \varnothing$ **then** return $\varnothing$
  $A \longleftarrow Nb(q)$
  $B \longleftarrow Nb(R[0])$
  if $A \cap B \neq \varnothing$ **then** return $\varnothing$
  else
    for $a \in A \backslash \{R[0]\}$ **do**
      if $Nb(a) \neq B$ **then** return $\varnothing$
    for $b \in B \backslash \{q\}$ **do**
      if $Nb(b) \neq A$ **then** return $\varnothing$
  return tuple (A, B)

sequence of node-sides c is both a valid internal chain and ends with r. Therefore (L, R) cannot be connected.

Proposition 24 (Validity of Algorithm 1). *This algorithm detects all balanced recombination bundles, and rejects all unbalanced recombination bundles.*

Proof. *Suppose q is involved in a balanced recombination bundle (L, R). W.l.o.g. suppose that $q \in L$. Due to the complete matching property, the set $Nb(q)$ in the algorithm is guaranteed to be equal to R. Due to the completeness property, the set $Nb(R[0])$ in the algorithm is guaranteed to be equal to L. It is evident that the algorithm directly verifies complete matching and acyclicity.*

Suppose otherwise. Assuming we have eliminated all tips, which can be done in $\mathcal{O}(|V|)$ time, Lemma 20 proves that q is involved in an unbalanced bundle B. If B fails acyclicity but not complete matching, then checking that $A \cap B = \varnothing$ will correctly detect that $L \cap R \neq \varnothing$.

Otherwise, suppose that B fails complete matching. Suppose first that $Nb(q) \subset R$. We assert that $\exists a \in Nb(q)$ such that $\exists b \in Nb(a)$ such that $\exists c \in Nb(b)$ such that $c \notin Nb(q)$. This event will be detected by the second loop of the algorithm. This follows from the connectedness of B and Lemma 23.

Suppose otherwise that $Nb(q) = R$ but $\exists r \in R$ such that $Nb(r) \subset L$. Let $c \in L \backslash Nb(r)$. By matching, $\exists r' \in R$ such that $r' \in Nb(c)$. Therefore $Nb(r)$ and $Nb(r')$ will be found to be unequal in the first loop of the algorithm.

Suppose otherwise that $Nb(q) = R$, $Nb(r) = L \, \forall r \in R$, but $\exists \ell \in L$ such that $Nb(\ell) \subset R$. Then we will find in the second loop that $Nb(\ell) \neq Nb(q)$.

Proposition 25 (Speed of Algorithm 1). *We can identify all balanced recombination bundles, all unbalanced bundles and all tips in $\mathcal{O}(|E| + |V|)$ time.*

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

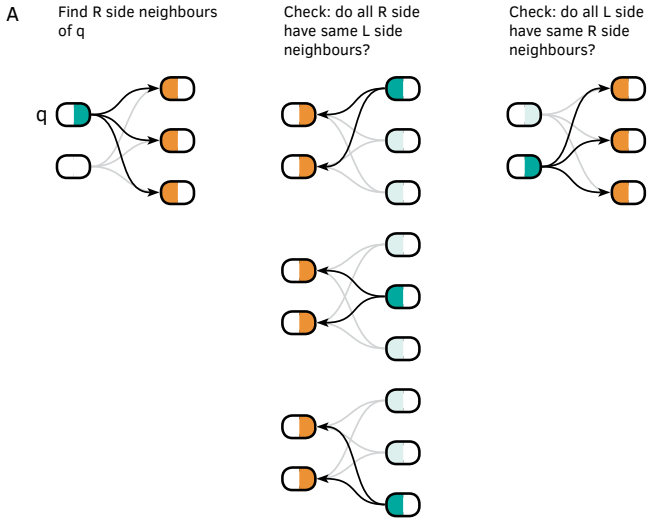

Fig. 16. Illustration of Algorithm 1 returning a positive result

Proof. We depend on a neighbor index giving us $\mathcal{O}(|Nb(n)|)$ *iteration across neighbors of a node-side* n*.*

We begin by looping over all node-sides and identifying all tips, which is achieved in $\mathcal{O}(|V|)$ *time. We then loop again over all remaining node-sides. At each node-side* q*, we run the function describe above, which, if* q *is involved in a balanced recombination bundle, will return the bundle* $B = \overline{W}(q)$*. It is evident that this function runs in* $\mathcal{O}(|E_B|)$ *time, seeing as it loops over each edge of* B *twice—once from each side—each time making an* $\mathcal{O}(1)$ *set inclusion query. After* B *is built, all nodes are marked such that they are skipped when they are encountered in the global loop. This gives overall* $\mathcal{O}(|E_B| + |V_B|)$ *exploration of* B*.*

If q *is involved in an unbalanced bundle* $B = \overline{W}(q)$*, this fact is detected by the same function in* $O(|E_B|)$ *time. In this case, we can find all nodes of* B *by performing a breadth-first search. Examination of the* W*-function will convince the reader that a breadth-first search will find all node-sides of* B *in* $\mathcal{O}(|E_B| + |V_B|)$ *time. We follow the same procedure of marking all these node-sides to be skipped in the global loop.*

This proves that, after eliminating tips in $\mathcal{O}(|V|)$ *time, we can build the set* $\mathbb{B}$ *of all non-isomorphic bundles* B*, and decide whether they are balanced recombination bundles, in time proportional to* $\sum_{B\in\mathbb{B}} |E_B| + |V_B|$*. But Lemma 20 and Corollary 22 tell us that* $V = \{v : v \text{ is a tip}\} \cup \bigcup_{B\in\mathbb{B}} V_B$*, and that all elements of this union of node-sides are disjoint. Furthermore, due to the*

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

matching property of bundles, $E = \bigcup_{B \in \mathbb{B}} E_B$, and all elements of this union of edges are disjoint. Therefore, our method is $\mathcal{O}(|V| + |E|)$.

5.3 Bundles and Snarl Boundaries

Definition 26. *Given a "boundary" node-side $b = s$ or t of a snarl (S, s, t), we call the tuple $(b, Nb(b))$ a snarl comb. A snarl comb is called proper if $\forall n \in Nb(b), Nb(n) = \{b\}$ and $b \notin Nb(b)$.*

It is easy to verify that a *proper snarl comb* is a balanced recombination bundle. It is also easy to see that an improper snarl comb is, according to set inclusion of tuples, a proper subset of a unique bundle.

Proposition 27 (Bundles do not cross snarl boundaries). *Let (S, s, t) be a snarl. Suppose that $B = (L, R)$ is a bundle. Then either all node-sides involved in B are members of S, or no node-side involved in B is a member of S.*

Proof. Suppose that there exists a bundle $B = (L, R)$ with node-sides both within S and not within S. Let x, y be involved in B, with $x \in S$, $y \notin S$. W.l.o.g., suppose $x \in L$, $y \in R$. This implies that there exists an internal chain $p = x \to \cdots \to y$. But then this implies that there exists $a \in S, b \notin S$ such that $a \in Nb(b)$, which would allow us to use the edge $a \to b$ to create a path violating the 2-node separability of S

5.4 Defining Sites using Bundles

Definition 28. *An ordered pair (B_1, B_2) of balanced recombination bundles is compatible if either*

1. *$\forall x \in R_1, \hat{x} \in L_2$, and $\forall y \in L_2, \hat{y} \in R_1$*
2. *$\exists$ a bijection $f : L_1 \longrightarrow R_2$ such that $\forall x \in R_1$, there exists a unique path $p(x)$ from $x \to \cdots \to f(x)$, and all paths $p(x)$ are disjoint.*

Definition 29. *If two recombination bundles are compatible, we define the set $p(x)$ to be a bundled simple site P.*

Claim. Consider a bundled simple site P in a graph G, lying between compatible balanced recombination bundles B_1, B_2. The set of all paths in G which contain paths $p \in P$ as contiguous subpaths is isomorphic to the set-theoretic product $P(L_1) \times P \times P(R_2)$ consisting of the three sets

1. $P(L_1) := \{\text{paths in } G \backslash S \text{ terminating in } x, \text{ for some } x \in L_1\}$
2. P
3. $P(R_2) := \{\text{paths in } G \backslash S \text{ beginning with } y, \text{ for some } y \in R_2\}$

under the function mapping $p_1 \in P(L_1), p \in P, p_2 \in P(R_2)$ to their concatenation.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

We will call a balanced recombination bundle $B = (L, R)$ *trivial* if both L and R are singleton sets.

Definition 30. *An ultrabubble (U, s, t) is a generalized simple bubble if*

1. *$(\{s\}, Nb(s))$ and $(Nb(t), \{t\})$ are balanced recombination bundles*
2. *The set of all non-trivial balanced recombination bundles admits a linear ordering $X \to B_1 \to \ldots B_k \to Y$ such that X and Y are either of $(\{s\}, Nb(s))$ and $(Nb(t), \{t\})$, X is compatible with B_1, every B_i is compatible with B_{i+1}, and B_k is compatible with Y*

Definition 31. *An ultrabubble U is decomposable into nested generalized sites if either:*

1. *It is a generalized simple bubble*
2. *When each ultrabubble (V, u, v) nested in U which is a decomposable into nested generalized sites is replaced with a single edge spanning u and v, then U is a generalized simple bubble*

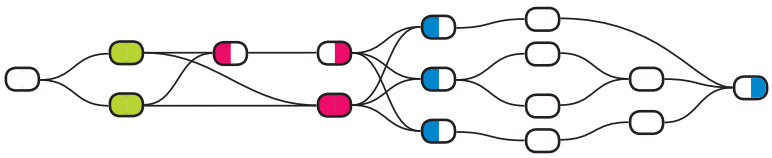

Fig. 17. An ultrabubble decomposable into nested generalized sites; some sites marked

We sketch a linear-time method of building sites from a tree diagram of nested ultrabubbles. We run Algorithms 2 and 3 starting at bottom-level nested ultrabubbles. If ultrabubble has all nontrivial balanced recombination bundles paired, then, when we evaluate the ultrabubble containing it, we represent it as a single edge from its source to sink.

In Algorithm 3, which follows below, we refer to the individual sets of node-sides forming the tuples (L, R) of a bundle as bundle-sides.

5.5 Bundles Containing Deletions

Our bundles—and therefore our sites—fail to detect the graph motifs formed by deletions spanning otherwise well-behaved variants. We define a special, well-behaved subclass of unbalanced bundle to address this.

Definition 32. *A deletion bundle-pair is a tuple (L_A, R_A, L_B, R_B) such that*

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Algorithm 2: Finding spans connecting bundles

Data: Ultrabubble U, and set $\mathbb{B}$ of balanced recombination bundles
Result: Set $\mathbb{P}$ of spans of unbranching sequence in U
begin
  $\mathbb{T} \longleftarrow$ vector of all trivial bundles in $\mathbb{B}$
  $N_{\mathbb{T}} \longleftarrow$ map $(N \to \mathbb{T})$ of node-sides to trivial bundles which contain them
  for *each trivial bundle* $t = (\{t_l\}, \{t_r\}) \in \mathbb{T}$ **do**
    if $N_{\mathbb{T}}[\hat{t_r}]$ *is found* **then**
      $(\{u_l\}, \{u_r\}) \longleftarrow N_{\mathbb{T}}[\hat{t_r}]$
      replace $(\{u_l\}, \{u_r\})$ in $\mathbb{T}$ with $(\{t_l\}, \{u_r\})$
      flag $(\{t_l\}, \{t_r\})$ as having been right-extended
  $\mathbb{P} \longleftarrow \varnothing$
  for $t = (\{t_l\}, \{t_r\}) \in \mathbb{T}$ **do**
    if t *not flagged as having been right-extended* **then** $\mathbb{P} \longleftarrow \mathbb{P} \cup \{t\}$
  return $\mathbb{P}$

Algorithm 3: Finding compatible bundles

Data: Ultrabubble U, set $\mathbb{P}$ of node-side tuples containing endpoints of spanning segments, and set $\mathbb{B}$ of balanced recombination bundles
Result: Set $\mathbb{C}$ of all compatible pairs of bundle-sides
begin
  $N_{\mathbb{P}} \longleftarrow$ map $(N \to \mathbb{P})$ of node-sides of $p \in \mathbb{P}$ to elements of $\mathbb{P}$
  $N_{\mathbb{B}} \longleftarrow$ map $(N \to \mathbb{B})$ of node-sides to bundles-sides of nontrivial R-bundles
  $\mathbb{C} \longleftarrow \varnothing$
  for *R-bundle side* $X \in \mathbb{B}$ **do**
    $x \longleftarrow X[0]$
    R-bundle side $X_{opposite} \longleftarrow \varnothing$, $Y \longleftarrow \varnothing$
    if $\hat{x}$ *found in* $N_{\mathbb{B}}$ **then**
      $X_{opposite} \longleftarrow N_{\mathbb{B}}[\hat{x}]$
      $Y \longleftarrow \{\hat{x}\}$
    else if $\hat{x}$ *found in* $N_{\mathbb{P}}$ **then**
      $y \longleftarrow$ node-side of $N_{\mathbb{P}}[\hat{x}]$ which isn't $\hat{x}$
      $X_{opposite} \longleftarrow N_{\mathbb{B}}[\hat{y}]$
      $Y \longleftarrow \{\hat{y}\}$
    if $X_{opposite} \neq \varnothing$ *and* $|X_{opposite}| = |X|$ **then**
      for $a \in X \backslash x$ **do**
        if $\hat{x}$ *found in* $N_{\mathbb{B}}$ **then** $Y \longleftarrow Y \cup \{\hat{x}\}$
        else if $\hat{x}$ *found in* $N_{\mathbb{P}}$ **then**
          $y \longleftarrow$ node-side of $N_{\mathbb{P}}[\hat{x}]$ which isn't $\hat{x}$
          $Y \longleftarrow Y \cup \{\hat{y}\}$
      if $Y = X_{opposite}$ **then**
        $\mathbb{C} \longleftarrow \mathbb{C} \cup \{(X, X_{opposite})\}$
  return $\mathbb{C}$

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

1. $\forall \ell \in L_A, \forall r \in R_A, \{\ell, r\} \in E$
2. $\forall \ell \in L_A, \forall r \in R_B, \{\ell, r\} \in E$
3. $\forall \ell \in L_B, \forall r \in R_B, \{\ell, r\} \in E$
4. $\exists$ *no other edge involving any node-side* $n \in L_A, L_B, R_A$ *or* R_B

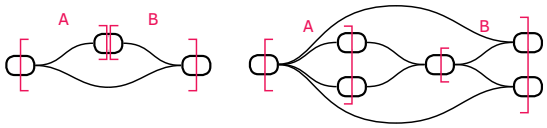

Fig. 18. Two examples of deletion bundle-pairs

These structures occur when two balanced recombination bundles on either side of some span of graph are bridged by deletions. It remains necessary to check that there is graph structure joining the nodes of R_A to L_B for this to be the case.

Algorithm 4 below will detect deletion bundle pairs from among the set of unbalanced bundles in linear time.

Proposition 33. *Given a set of acyclic unbalanced bundles, this algorithm finds those among then which are deletion bundle pairs.*

Proof. Suppose that q *is involved in a deletion bundle pair* (L_A, R_A, L_B, R_B). *W.l.o.g, either* $q \in L_A$ *or* $q \in L_B$.

Suppose first that $q \in L_B$*: In this case,* $Nb(q) = R_B$. *We then know that* $\forall a \in R_B$, $Nb(a) = L_A \cup L_B$. *This will trigger the condition* $L_2 = \varnothing$. *The elements of* $a \in L_1$ *will segregate into precisely two groups: one such that* $Nb(a) = R_B$*—the elements* $a \in L_B$, *and another group such that* $Nb(a) = R_A \cup R_B$*—the elements* $a \in L_A$. *If these conditions are fulfilled, we then build* R_A *and* R_B. *It remains to verify that* $\forall b \in R_A, Nb(b) = L_A$, *and* $\forall b \in R_B, Nb(b) = L_A \cup L_B$.

Suppose otherwise that $q \in L_A$*: In this case,* $Nb(q) = R_A \cup R_B$. *This will trigger the condition* $L_2 \neq \varnothing$ *since the elements* $b \in Nb(q)$ *will segregate into two groups:* R_A, *where if* $b \in R_A, Nb(b) = L_A$ *and* R_B, *where if* $b \in R_B, Nb(b) = L_A \cup L_B$. *If this condition is met, then it remains to check that* $\forall a \in L_A, Nb(a) = R_A \cup R_B$ *and* $\forall a \in L_B, Nb(a) = R_B$.

Suppose otherwise that q *is not involved in a deletion bundle pair. Suppose that Algorithm 4 does not fail, returning* $\varnothing$. *There are two possibilities then for the nature of the unbalanced bundle* (L, R) *for which* $q \in L$.

First, suppose the condition $L_2 = \varnothing$ *was triggered. The* $\exists q \in L$ *such that, where* $L' := \{l \in L \mid l \in Nb(a)$ *for some* $a \in Nb(q)\}$, $Nb(\ell) \subseteq Nb(q)\ \forall \ell \in L'$. *Then by Lemma 23,* $Nb(\ell) \geq Nb(q) \forall \ell \in L$. *Therefore it must be that* $Nb(q) = R$. *Furthermore, to pass the search for* R_A, *there must* $\exists R_A$ *such that if* $\ell \in L$

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Algorithm 4: Deletion bundle pair finding

Data: Node-side q known to be in an acyclic unbalanced bundle
Result: Deletion bundle-pair containing q if it is in a deletion bundle-pair, $\varnothing$ otherwise

begin
- $L_A, R_A, L_B, R_B \longleftarrow \varnothing$
- $R_{temp} \longleftarrow Nb(q)$
- $L_1 \longleftarrow Nb(R_{temp}[0])$, $L_2 \longleftarrow \varnothing$
- **for** $a \in R_{temp} \backslash \{R_{temp}[0]\}$ **do**
  - **if** $Nb(a) \neq L_1$ **then**
    - **if** $L_2 = \varnothing$ **then** $L_2 \longleftarrow Nb(a)$
    - **else if** $Nb(a) \neq L_2$ **then** return $\varnothing$
- **if** $L_2 \neq \varnothing$ **then**
  - **if** $L_2 \subset L_1$ **then** $L_A \longleftarrow L_2$, and $L_B \longleftarrow L_1 \backslash L_2$
  - **else if** $L_1 \subset L_2$ **then** $L_A \longleftarrow L_1$, and $L_B \longleftarrow L_2 \backslash L_1$
  - **else** return $\varnothing$
  - $R_{temp} \longleftarrow Nb(L_B[0])$
  - **for** $a \in L_B \backslash \{L_B[0]\}$ **do**
    - **if** $Nb(a) \neq R_{temp}$ **then** return $\varnothing$
  - $R_{temp} \longleftarrow Nb(L_A[0])$
  - **for** $a \in L_A \backslash \{L_A[0]\}$ **do**
    - **if** $Nb(a) \neq R_{temp}$ **then** return $\varnothing$
  - **if** $Nb(L_B[0]) \subset Nb(L_A[0])$ **then**
    - $R_B \longleftarrow Nb(L_B[0])$
    - $R_A \longleftarrow Nb(L_A[0]) \backslash Nb(L_B[0])$
  - **else** return $\varnothing$
- **else**
  - $R_B \longleftarrow Nb(q)$
  - **for** $a \in L_1 \backslash \{q\}$ **do**
    - **if** $Nb(a) \neq R_B$ **then**
      - **if** $R_A = \varnothing$ **then**
        - **if** $R_B \not\subset Nb(a)$ **then** return $\varnothing$
        - **else** $R_A \longleftarrow Nb(a) \backslash R_B$
      - **else if** $Nb(a) \neq R_A \cup R_B$ **then** return $\varnothing$
  - **if** $R_A = \varnothing$ **then** return $\varnothing$
  - $L_A \longleftarrow Nb(R_A[0])$
  - **for** $a \in R_A \backslash R_A[0]$ **do**
    - **if** $Nb(a) \neq L_A$ **then** return $\varnothing$
  - $L_{temp} \longleftarrow Nb(R_B[0])$
  - **for** $a \in R_B \backslash R_B[0]$ **do**
    - **if** $Nb(a) \neq L_{temp}$ **then** return $\varnothing$
  - **if** $L_A \subset L_{temp}$ **then** $L_B \longleftarrow L_{temp} \backslash L_A$
  - **else** return $\varnothing$
- return tuple (L_A, R_A, L_B, R_B)

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

and $Nb(\ell) \neq R$*, then* $Nb(\ell) = R_A$*. Furthermore, to pass the conditions of the subsequent two loops, it must be that* $\forall r \in R \backslash R_A$*, all* $Nb(r)$ *are the same, and* $\forall r' \in R_A$*, all* $Nb(r')$ *are the same. Furthermore, to pass the last condition checked, must be that* $Nb(r')$ *from the latter group* $\subset Nb(r)$*. And since* $L_A := \{\ell \in L \mid Nb(\ell) = R\}$ *and* $L_B := \{\ell \in L \mid Nb(\ell) = R_B\}$ *are such that* $L_A \cap L_B = \varnothing, L_A \cup L_B = L$*, these conditions all together ensure that* (L, R) *is a deletion bundle pair.*

Otherwise, $Nb(q)$ *segregates into two disjoint subsets* $R_A := \{r \in Nb(q) | Nb(r) = L_A\}$*,* $R_B := \{r \in Nb(q) \mid Nb(r) = L_A \cup L_B$ *for some* $L_A, L_B \subset L$ *such that* $L_A \cap L_B = \varnothing$*. To pass further conditions, it is necessary that* $\forall \ell \in L_B, Nb(\ell) = R_B$ *and* $\forall \ell \in L_A, Nb(\ell) = R_A \cup R_B$*. It remains to show that* $L_A \cup L_B = L$ *and* $R_A \cup R_B = R$*, these can be proven by application of Lemma 23. Therefore in this case, it must also be that* (L, R) *is a deletion bundle pair.*

Proposition 34. *This algorithm finds deletion bundles in* $\mathcal{O}(|E| + |V|)$ *time.*

Proof. Note that a deletion bundle-pair is a special type of unbalanced bundle. Therefore, if, given an unbalanced bundle B*, we can check whether it is a deletion bundle-pair in* $\mathcal{O}(|E_B| + |V_B|)$ *time, by the arguments of Proposition 21, we can find all deletion bundle-pairs in* $\mathcal{O}(|E| + |V|)$ *time.*

Inspection of the algorithm shows that, like the algorithm for identifying balanced recombination bundles, it performs two $\mathcal{O}(1)$ *set-inclusion queries per edge, making it* $\mathcal{O}(|E_B|)$ *overall.*

6 Discussion

Graph formalism has the potential to revolutionize the discourse on genetic variations by creating a model and lexicon that more fully embraces the complexity of sequence change. This is vital: the current linear genome model of a reference sequence interval and alternates is insufficient. It fails to express nested variation and can not properly describe information about the breakpoints that comprise structural variations.

The introduction, in order, of bubbles, superbubbles, ultrabubbles and snarls progressively generalizes the concept of a genetic site to accommodate more general types of variation using progressively more general graph types. In this paper we both review and build on these developments, showing how the recently introduced ultrabubbles can be furthered sub-classified using concepts from circuit theory. This expands the simple notion of proper nesting described in the original ultrabubble paper. Furthermore, we describe how we can extend the theory of ultrabubbles by generalizing ultrabubble boundaries to another sort of boundary structure—the bundle—which allows us to describe regions where variants are packed too closely to be segregated into separate ultrabubbles.

Our methods are powerful in decomposing dense collections of nested or closely packed variation into meaningful genetic sites. We anticipate that these structures will become increasingly common in the analysis of variation using graph methods, as sequencing datasets containing variation from increasing numbers of individuals become available.

bioRxiv preprint doi: https://doi.org/10.1101/125542; this version posted April 13, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Acknowledgements

Y.R. is supported by a Howard Hughes Medical Institute Medical Research Fellowship. This work was also supported by the National Human Genome Research Institute of the National Institutes of Health under Award Number 5U54HG007990 and grants from the W.M. Keck foundation and the Simons Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We thank Wolfgang Beyer for his visualizations of 1000 Genomes data in a variation graph.

References

1. Danecek, P., Auton, A., Abecasis, G., Albers, C.A., Banks, E., DePristo, M.A., Handsaker, R.E., Lunter, G., Marth, G.T., Sherry, S.T., et al.: The variant call format and vcftools. Bioinformatics **27** (2011) 2156–2158
2. 1000 Genomes Project Consortium, et al.: A global reference for human genetic variation. Nature **526** (2015) 68–74
3. Sudmant, P.H., Rausch, T., Gardner, E.J., Handsaker, R.E., Abyzov, A., Huddleston, J., Zhang, Y., Ye, K., Jun, G., Fritz, M.H.Y., et al.: An integrated map of structural variation in 2,504 human genomes. Nature **526** (2015) 75–81
4. Novak, A.M., Hickey, G., Garrison, E., Blum, S., Connelly, A., Dilthey, A., Eizenga, J., Elmohamed, M.A.S., Guthrie, S., Kahles, A., Keenan, S., Kelleher, J., Kural, D., Li, H., Lin, M.F., Miga, K., Ouyang, N., Rakocevic, G., Smuga-Otto, M., Zaranek, A.W., Durbin, R., McVean, G., Haussler, D., Paten, B.: Genome graphs. bioRxiv (2017)
5. Beyer, W.: Sequence tube maps. https://github.com/wolfib/sequenceTubeMap (2016)
6. Medvedev, P., Brudno, M.: Maximum likelihood genome assembly. Journal of computational Biology **16** (2009) 1101–1116
7. Paten, B., Novak, A.M., Garrison, E., Hickey, G.: Superbubbles, ultrabubbles and cacti. bioRxiv (2017)
8. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de bruijn graphs. Genome research **18** (2008) 821–829
9. Onodera, T., Sadakane, K., Shibuya, T. In: Detecting Superbubbles in Assembly Graphs. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 338–348
10. Brankovic, L., Iliopoulos, C.S., Kundu, R., Mohamed, M., Pissis, S.P., Vayani, F.: Linear-time superbubble identification algorithm for genome assembly. Theoretical Computer Science **609, Part 2** (2016) 374 – 383
11. Sung, W.K., Sadakane, K., Shibuya, T., Belorkar, A., Pyrogova, I.: An o(m log m)-time algorithm for detecting superbubbles. IEEE/ACM Trans. Comput. Biol. Bioinformatics **12** (2015) 770–777
12. Valdes, J., Tarjan, R.E., Lawler, E.L.: The recognition of series parallel digraphs. SIAM Journal on Computing **11** (1982) 298–313
13. Duffin, R.: Topology of series-parallel networks. Journal of Mathematical Analysis and Applications **10** (1965) 303 – 318