Identification, genotyping and representation of structural variants in pangenomes

Sandra Romain

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : Informatique

Par
Sandra ROMAIN

Identification, génotypage et représentation des variants structuraux dans les pangénomes

Thèse présentée et soutenue à Rennes, le 8 novembre 2024
Unité de recherche : IRISA, Centre Inria de l'Université de Rennes – Équipe GenScale

Rapporteurs avant soutenance :

Birte KEHR	Professeure, University of Regensburg, Leibniz Institute of Immunotherapy, Regensburg (Allemagne)
François SABOT	Directeur de recherche, IRD Montpellier

Composition du Jury :

Examinateurs :	Birte KEHR	Professeure, University of Regensburg, Leibniz Institute of Immunotherapy, Regensburg (Allemagne)
	François SABOT	Directeur de recherche, IRD Montpellier
	Pierre PETERLONGO	Directeur de recherche, INRIA/IRISA Rennes
	Sèverine BERARD	Maîtresse de conférence, Université de Montpellier
Dir. de thèse :	Claire LEMAITRE	Directrice de recherche, INRIA/IRISA Rennes
Co-enc. de thèse :	Fabrice LEGEAI	Ingénieur de recherche, INRAE - IGEPP Rennes

Acknowledgements

First of all, I would like to thank Birte Kehr and François Sabot for reviewing my PhD work and their help in improving this manuscript. I would also like to thank Sèverine Berard and Pierre Peterlongo for accepting to be part of my jury. Thank you all for all your remarks and the interest you took in my work.

I would also like to give very special thanks to my awesome supervisors Claire Lemaitre and Fabrice Legeai. Thank you Claire for giving me the opportunity to do my M2 internship with you and take a dive into the world of sequence algorithm and structural variants, and then trusting me on this PhD project. I feel really lucky to have had such a dedicated and available supervisor. Thank you Fabrice for taking the role of my PhD co-supervisor, your expertise helped me a lot in the analysis of our *Coenonympha* genomes. It was a real pleasure to work with you both and benefit from your time and precious advices, I learned a lot by your side. Thank you for accompanying me along these 3-4 years and helping me grow in my work and research.

Also, big thanks to the Divalps collaborators and the interest you have shown in my work, and especially to Thibaut for your help on the *Coenonympha* SNP analysis, as well as to Laurence and Mathieu for your advices and time on the *Coenonympha* paper.

Thanks to the GenScale team and all of the Symbiose colleagues for your warm welcome and all the nice coffee breaks, but also the game and movie nights. Thank you to the Science en Cour[t]s team (Baptiste, Kerian, Khodor, Roland), I had a blast producing our amazing 'Patatogène' movie. A special mention to my dear office mates, Kerian (there are too many things to fit here, but thank you for Tsuki) and Francesca (I'll try my best to be a *maladetta monella* ;)).

Finally, big thanks to my friends, family and partner, who have had to put up with me during the sometimes intense times of this PhD.

Résumé En Français

Introduction

Les variants génomiques sont un facteur majeur de la diversité du vivant. Parmi eux, les variants de structure (SVs) sont des variants d'au moins 50 paires de bases (pb), pouvant être classifiés en cinq types de base : délétion, insertion, inversion, duplication et translocation. L'analyse et la comparaison des SVs entre individus dans une espèce se fait en deux temps : la détection et le génotypage. La détection des SVs consiste à former une liste de SVs bien décrits (par exemple par leur type, leur position, leur taille). Le génotypage consiste alors à comptabiliser les allèles des SVs connus dans des individus re-séquencés. La détection et le génotypages des SVs sont traditionnellement réalisés par alignement de lectures sur un génome de référence. L'avènement des technologies de séquençage à longues lectures a révolutionné la détection des SVs dans les génomes, en améliorant l'alignement des lectures dans les régions génomiques riches en répétitions, dans lesquelles plus de la moitié des SVs se situent. Cela a remis en lumière leur contribution à la diversité génétique, et leur implication dans la variabilité phénotypique, l'adaptation et l'évolution des espèces, et certaines maladies cliniques. La précision du génotypage est cruciale dans la découverte de ces associations entre SVs et phénotypes.

Les nouvelles structures de données que sont les graphes de variation et graphes de pangénomes (2018-2023) apportent plein de promesses pour l'analyse de la diversité génétique, et notamment pour la détection, le génotypage et l'analyse des SVs. Ces deux types de graphes sont très similaires en termes de structure, et représentent les variants sous forme de bulles, cependant, ils se construisent de manière différente. Le graphe de variation est construit à partir d'un génome de référence et d'une liste de variants. Le graphe de pangénome est construit par alignement de génomes complets à partir d'une collection de génomes. Leur représentation inhérente de la variabilité de séquence d'une espèce rend l'alignement des lectures moins sujette au biais vers la référence, et permet ainsi une amélioration de la découverte et du génotypage de variants de toutes tailles et de tous types. Plusieurs méthodes de génotypage de SVs utilisant les graphes de variation et de pangénomes on été développées, mais toutes utilisent des lectures courtes.

Cette thèse s'inscrit dans le cadre du projet ANR DIVALPS, qui s'intéresse à l'histoire évolutive particulière d'un complexe de quatre espèces de papillons, les *Coenonympha* alpins. De précédents travaux des collaborateurs du projet suggèrent fortement l'occurrence d'un événement d'hybridation entre *C. arcania* et *C. gardetta*, suivi de l'isolement génétique et géographique de deux lignées hybrides,

C. darwiniana et *C. cephalidarwiniana*. L'un des objectifs du projet DIVALPS est d'explorer les facteurs génétiques et génomiques impliqués dans la spéciation de ces lignées hybrides et leur adaptation à différents habitats. Les inversions, qui sont des réarrangements génomiques réduisant localement le taux de recombinaison effectif à l'état hétérozygote et pouvant mener à la capture de combinaisons alléliques spécifiques et à leur maintien dans les populations, représentent une piste de recherche intéressante. Les grandes inversions, en particulier, ont une plus grande probabilité de capturer des associations alléliques pouvant impacter l'évolution des espèces (p.e. associations promouvant l'adaptation à certaines conditions écologiques). Cependant, la présence fréquente de motifs répétés à leur extrémité, ainsi que la divergence de séquence entre génomes, rendent les grandes inversions entre espèces plus difficiles à caractériser avec précision à partir de l'alignement de lectures courtes ou longues. De récentes méthodes de détection des SVs s'appuyant sur de l'alignement de génomes complets ont démontré pouvoir surpasser ces limites.

Cette thèse avait pour objectifs de développer de nouvelles méthodes de détection et de génotypage des SVs en employant ces nouveaux objets que sont les graphes de variation et de pangénomes. Le placement de cette thèse dans le projet d'analyse des génomes *Coenonympha* a focalisé ces objectifs sur un contexte de génome non modèles et de comparaison de génomes complets, et apporté une dimension d'évaluation des méthodes existantes dans ce contexte.

Contributions de la thèse

Un outil basé sur les graphes de variations pour génotyper rapidement et précisément les variants de structures avec lectures longues

J'ai débuté ma thèse en travaillant sur l'utilisation du modèle de graphe de variation dans le cadre du génotypage de SVs avec lectures longues. Les données de séquençage en lectures longues des papillons *Coenonympha* n'étaient pas encore disponibles à ce moment-là, mais les graphes semblaient déjà être une approche intéressante pour la comparaison des génomes *Coenonympha*, qui ne nécessitait pas de choisir un génome de référence entre les deux espèces parentales. Ce travail m'a donc permis d'explorer comment les SVs pouvaient être représentés dans de tels graphes. De plus, toutes les méthodes de l'état de l'art dédiées au génotypage des SVs avec des lectures longues sont limitées soit par leur représentation inégale des différents allèles des SVs, qui entraîne un biais vers la référence, soit par leur représentation

linéaire des allèles, qui entrave le génotypage des SVs proches et chevauchants. L'utilisation du graphe de variation était donc une piste prometteuse pour résoudre ces limites identifiée dans l'état de l'art, et en particulier la limite du génotypage de SVs proches identifiée dans SVJedi, un outil développé dans mon équipe de recherche.

J'ai développé et implémenté une nouvelle méthode, SVJedi-graph (Romain and Lemaitre, 2023), qui construit un graphe de variation à partir d'un génome de référence et d'une liste de SVs connus. Cette méthode utilise un mappeur de séquences sur graphe de l'état de l'art pour aligner les longues lectures sur le graphe de variation, et estime le génotype le plus probable pour chaque SV à partir de la profondeur des lectures cartographiées sur les arcs représentant les différents allèles dans le graphe. En appliquant SVJedi-graph sur des ensembles simulés de délétions proches et chevauchantes, j'ai démontré que ce modèle de graphe permet de surmonter le problème de biais vers la référence tout en maintenant une grande précision de génotypage, quelle-que soit la proximité des SVs. Sur le jeu de données de référence humain HG002 produit par le consortium *Genome in a Bottle*, SVJedi-graph a obtenu les meilleures performances parmi les génotypeurs de SVs utilisant des longues lectures, réussisant à prédire en moins de 30 minutes le génotype de 99,5 % des SVs du jeu de données avec une précision de 95 %.

Découverte de douze grandes inversions dans les génomes des papillons alpins *Coenonympha*

L'un des objectifs du projet DIVALPS est de mettre en lumière les facteurs génétiques et génomiques impliqués dans l'histoire évolutive particulière du complexe d'espèces de papillons alpins *Coenonympha*. Dans ce but, les quatre espèces *Coenonympha* de ce complexe ont été séquencées, par lectures longues PacBio CLR pour un individu par espèce, et par lectures courtes Illumina pour 9 à 19 individus par espèce. Le jeu de données de lectures longues a permi la production et la publication du premier génome assemblé de *Coenonympha arcania* (Legeai et al., 2024), à laquelle j'ai participé, ainsi que des premiers assemblages des trois autres espèces. L'accès à ces génomes de haute qualité a été un élément crucial dans la réalisation de cette partie de ma thèse, au cours de laquelle j'ai cherché à comparer les génomes des quatres espèces de *Coenonympha* pour détecter des SVs entre les espèces. J'ai pu tester différents outils basés sur alignement de lectures longues ou sur alignement de génomes complets. Une grosse partie de la comparaison entre les génomes de *Coenonympha* s'est concentrée sur la découverte de grandes inversions.

En utilisant une méthode basée sur alignement de génomes complets, j'ai découvert et caractérisé 12 grandes inversions ($\geq$ 100 kbp) entre les quatre espèces, dont 2 on été détectées à l'état hétérozygote dans les génomes assemblés de *C.*

cephalidarwiniana. L'analyse complémentaire des statistiques génétiques de données populationnelles à l'intérieur de ces inversions, réalisée par nos collaborateurs sur ce projet, suggère qu'au moins 3 inversions sont présentes à l'état hétérozygote dans certaines populations de *C. cephalidarwiniana*, et que 5 inversions contiennent des régions barrières au flux de gènes.

Analyse des motifs topologiques des inversions dans les graphes de pangénomes

Après avoir identifié les grandes inversions chez les *Coenonympha* alpins avec les approches classiques, j'ai voulu tester la possibilité de retrouver ces même grandes inversions dans les graphes de pangénomes. Le graphe de pangénome présente au moins deux avantages par rapport aux analyses basées sur génome de référence dans le cas d'étude des papillons *Coenonympha*. Premièrement, il permet de considérer les deux génomes d'espèces parentes sur un même pied d'égalité, plutôt que devoir choisir arbitrairement l'un ou l'autre en tant que génome de référence. Deuxièmement, il permet de comparer les inversions identifiées simultanément entre les quatre génomes plutôt que de devoir comparer les six paires de génomes, ce qui implique l'utilisation de plusieurs références et compléxifie la mise en commun des jeux d'inversions identifiées pour chaque paire.

J'ai remarqué que la détection d'inversions à partir de la topologie des graphes de pangénomes n'était pas immédiate avec les outils actuellement disponibles. Contrairement à d'autres types de variants (SNPs, délétions, insertions), les grands variants, et les inversions en particulier, sont plus difficiles à identifier à partir de la taille des chemins des bulles uniquement. En réponse à cela, j'ai développé une méthode et un outil, INVPG-annot, pour identifier les inversions parmi les bulles d'un graphe de pangénome. J'ai détecté très peu des inversions *Coenonympha* dans les graphes de pangénomes, ce qui m'a amené à concevoir plusieurs jeux de données simulées afin d'identifier les causes réelles de cette absence de détection des inversions. En testant mon outil sur des graphes de pangénomes construits par les quatre outils de l'état de l'art sur les différents jeux de données simulées, j'ai démontré que les outils diffèrent grandement dans leur façon et capacité à représenter précisément les inversions.

Conclusion

Les travaux de cette thèse ont mené à la mise à disposition de deux outils (SVJedi-graph et INVPG-annot), dont l'un a fait l'objet d'une publication dans le journal

Bioinformatics (Romain and Lemaitre, 2023) et dont l'autre a mené à l'écriture d'un papier prochainement soumis (Romain et al., prepb).

SVJedi-graph est le premier outil de génotypage de SVs avec lectures longues s'appuyant sur le modèle de graphe de variation. Cette nouvelle méthode a le potentiel d'améliorer significativement la précision de génotypage dans les régions riches en SVs.

En ce qui concerne la détection d'inversions entre génomes d'espèces différentes, aucune des différentes stratégies testées ne s'est révélée idéale. La stratégie basée sur génome de référence a fourni les résultats les plus complets (Romain et al., prepa), mais présente des inconvénients pour la comparaison de plus de deux génomes (traitement supplémentaire et conséquent des résultats). La stratégie basée sur graphes de pangénomes permettrait d'éviter ces inconvénients, mais souffre encore d'un manque d'outils pour identifier, analyser et exploiter correctement les grands variants dans ce type de graphe.

Contents

Acknowledgements i

Résumé en français iii

Introduction iii
Contributions de la thèse iv
Un outil basé sur les graphes de variations pour génotyper rapidement et précisément les variants de structures avec lectures longues iv
Découverte de douze grandes inversions dans les génomes des papillons alpins *Coenonympha* v
Analyse des motifs topologiques des inversions dans les graphes de pangénomes vi
Conclusion vi

Contents ix

List of Figures xiii

List of Tables xiv

I Introduction 1

1 Preamble 1
2 Structural variants in genomes 2
2.1 What is a genome? 2
2.2 What is a structural variant? 3
2.3 Types of structural variants 4

2.4 Impacts of structural variants 4
3 Methodological problems for variant analysis 6
3.1 Retrieving the sequence of a genome 6
3.2 Comparing genomes 8
3.3 Representing the genomic variability 11
4 The special case of inversions 13
4.1 Inversions impacts and dynamics in genomes 13
4.2 Introduction of the alpine *Coenonympha* butterfly species 14
5 Thesis objectives 17

II State of the Art 19

1 Structural variant discovery 20
1.1 From sequencing reads to genome assemblies 20
1.2 Methods for SV discovery from *de novo* assemblies 22
2 Structural variant genotyping with long reads 26
2.1 General strategy for structural variant genotyping 26
2.2 Two categories of long-read genotyping tools 27
2.3 Current limits of structural variant genotyping 29
3 Pangenome graphs: a new era for the characterization of SVs 30
3.1 Advantages of pangenome graphs over variation graphs 31
3.2 Pangenome graph construction 31
3.3 Structural variants in pangenome graphs 32
3.4 Current limits of SV characterization in pangenome graphs 33

III Long-read SV genotyping on a variation graph 35

1 Introduction 35
Paper: SVJedi-graph: improving the genotyping of close and overlapping structural variants with long reads using a variation graph 36

IV Inversion discovery in the *Coenonympha* species 47

1 Introduction 47

Paper: Characterization of large inversions to investigate hybrid speciation in the four species-complex of alpine *Coenonympha* butterfly 48

V Inversions in pangenome graphs 67

1 Introduction 67
Paper: Investigating the topological motifs of inversions in pangenome graphs 68

VI Discussion and Perspectives 87

Genotyping structural variants using graphs 87
A fast and accurate long-read SV genotyper on variation graph 87
Managing imprecise breakpoints 89
Adaptation of SVJedi-graph's method to pangenome graphs 90
Uncompleteness of structural variant genotyping evaluation 90
Inversion discovery: reference genome *versus* pangenome graph . 91
Reference-based SV discovery tools 91
Inversion annotation from pangenome graphs 92
Final thoughts on inter-specific inversion discovery 93

Bibliography 95

List of Figures

I Introduction 4

1 Basic types of structural variants 5
2 Example of a complex SV 5
3 Illustration of the short and long read mapping steps 10
4 Illustration of the VCF format 12
5 Illustration of a subset of variation graph 13
6 Example of the *Heliconius numata* P supergene 15
7 Morphology and distribution range of the four *Coenonympha* species 16

II State of the Art 20

8 SV discovery performances of short and long read sequencing 21
9 SV signatures used for SV discovery with short and long reads 22

List of Tables

II State of the Art 26

1 Comparison of state of the art assembly-based SV discovery tools 26
2 Main method differences between the long-read mapping-based SV genotyping tools 27

I Introduction

In this chapter

1 Preamble 1
2 Structural variants in genomes 2
2.1 What is a genome? 2
2.2 What is a structural variant? 3
2.3 Types of structural variants 4
2.4 Impacts of structural variants 4
3 Methodological problems for variant analysis 6
3.1 Retrieving the sequence of a genome 6
3.1.1 Genome sequencing 6
3.1.2 Genome assembly 7
3.2 Comparing genomes 8
3.2.1 From assemblies: whole-genome alignment 8
3.2.2 From reads: read mapping 9
3.2.3 Variant calling and genotyping 10
3.3 Representing the genomic variability 11
3.3.1 VCF and reference genome 11
3.3.2 Variation graphs 11
3.3.3 Pangenome graphs 13
4 The special case of inversions 13
4.1 Inversions impacts and dynamics in genomes 13
4.2 Introduction of the alpine *Coenonympha* butterfly species 14
5 Thesis objectives 17

1 Preamble

This chapter presents the biological aspects and motives behind structural variants and their characterization (Section 2), and introduces the problems raised surrounding their analysis (Section 3). Then, section 4 will focus on inversions and how this type of structural variant can impact genomes and organisms in more ways than other variant types, followed by the introduction of a key biological model of the thesis, the alpine *Coenonympha* butterflies.

2 Structural variants in genomes

DNA is the core component of life, carrying the genetic information for organism's development, growth and functions. It encodes the genome of an organism, which is present in all of its cells (with few exceptions like human red blood cells). Although all the cells of an organism inherit the same genome, which is a combination of half of each parental genome, some genomic variations can appear between somatic cells (*i.e.* non reproductive cells) during an organism's life. Whether these variations are caused by internal DNA-related molecular mechanisms or external factors (*e.g.* mutagens), they can alter the function of genes and impair the proper functioning of the cell. Depending on the genes affected, genomic variations can have an impact well beyond a single cell. For example, an accumulation of mutations in genes involved in cell-cycle control can lead to tumors and sometimes cancer.

This section introduces the concept and characteristics of genomes (2.1), needed to comprehensively define genomic variants and structural variants in particular (2.2), which are a category of genomic variants with heterogeneous sizes and types (2.3) that can impact genomes and organisms in multiple ways (2.4).

2.1 What is a genome?

DNA is a large molecule made of two chains - called *strands* - of smaller components, the *nucleotides* (nt). The *genome* designates the unique combination and order of nucleotides found in an organism's DNA, and can thus be transcribed as a *sequence* on a four character alphabet {A, C, G, T}.

Genomes can be found at different orders of size in nature. The size of a genome is generally said to be linked to the complexity of the organism; prokaryotes (*e.g.* bacteria) have typically the smallest genomes among living organisms with sizes ranging from 1 to 9 megabase pairs (Mbp) (Trevors, 1996), while eukaryotes (*e.g.* animals, plants and fungi) have genome sizes ranging from 2.3 Mbp to 148.8 gigabase pairs (Gbp) (Hidalgo et al., 2017). This genome can be fully contained in a single chromosome (*i.e.* a single DNA molecule) such as in bacteria, or spread across multiple chromosomes, which is the case for more complex organisms and larger genomes (*e.g.* animals and plants). For instance, the human genome is made of 3.1 Gbp distributed between 23 chromosomes.

In eukaryotic organisms, most cells contain two copies of each chromosome within an organelle called the nucleus, except for gametes, which contain only one copy. Such organisms are described as diploid. In such diploid organims, the two versions of a region that can exist at a particular locus are called alleles. Alleles can vary slightly in their sequence of DNA bases, leading to different traits. A locus is considered homozygous if it carries two copies of the same allele, and

heterozygous if it carries two different alleles. At the population level, alleles are distributed among individuals. A polymorphic locus can be bi-allelic if only two versions of the sequence exist, or multi-allelic if more than two alleles are present. However, some organisms have more than two sets of chromosomes, a condition known as polyploidy. This is common in plants and can also occur in some animals. Depending on the number of chromosome sets, these organisms are classified as triploid (3n, with three copies of each chromosome), tetraploid (4n), or hexaploid (6n).

2.2 What is a structural variant?

Genomic variants can be found at different scales. They can occur between somatic cells in a single individual, or occur in reproductive cells. Any genomic variation carried by reproductive cells can be transmitted to the progeny, in which case it will be carried by all cells of the new organism, thus introducing genomic variation between individuals in a population. Genomic variants are divided into several categories mainly depending on their size. The most common type of variation is Single-Nucleotide Polymorphism (SNP), with an average number of 4 million in human genomes, which accounts for 79 % of the detected variants in this species (Byrska-Bishop et al., 2022; Taylor et al., 2024). SNPs are small variations of 1 base pair (bp) that result from nucleotide substitution, caused by errors in the DNA replication mechanism or induced under the effect of external mutagens such as UV light.

On the other hand, structural variants (SVs) are genomic variants that affect at least 50 bp [1]. Although they are fewer in genomes compared to smaller variants, they can affect a larger portion of genomes. For instance, the human genome contains 25,000 to 35,000 SVs on average, which represents less than 1 % of the detected variants, but the bases covered by SVs account for 83 % of the total variant bases (Liao et al., 2023; Taylor et al., 2024). Collins et al. (2020) referenced and classified all SVs identified in the human genome from a panel of ∼15,000 individuals. Structural variants result from *genomic rearrangements* caused by double-stranded breaks of the DNA that are not correctly repaired by homologous recombination. These rearrangements introduce new sequence adjacencies in genomes where the DNA breaks occurred (called *breakpoints*). The different versions of the genome segments delimited by the breakpoints that can be found in a species are called the *alleles* of the SV. These genomic rearrangements can result in portions of the chromosomes being lost, duplicated, moved to another location and/or reoriented.

[1] The minimum length threshold of SVs is arbitrarily defined and may occasionally be found set to 30 bp in the literature, though the most commonly accepted one is 50 bp.

2.3 Types of structural variants

Structural variants are a variant category highly heterogeneous in size and type. The type of a structural variant corresponds to the type of the associated rearrangement, defined relatively to a chosen 'reference genome'. We can count five basic types, namely deletion, insertion, inversion, duplication and translocation, illustrated in Fig 1. As an example, a deletion describes a sequence s_d between two breakpoints being cut off the genome, creating a new adjacency between the sequences directly preceding and following s_d. The two possible alleles of this single deletion event correspond to either the presence of the deletion (*i.e.* allele carrying the new adjacency and missing s_d) or the absence of the deletion (*i.e.* allele carrying s_d, which is the allele of the 'reference genome'). Some SV types, like insertion and duplication, can be further characterized by the origin of the inserted sequence (*i.e.* cut and paste insertion, novel element insertion) or the relative position of the duplicated fragment (*i.e.* tandem duplication, interspersed duplication). There can be occurrences where multiple types of rearrangements are combined and form a complex SV with more than two breakpoints. An example of a complex SV is illustrated in Figure 2, where a deletion event and an inversion event both share a breakpoint with each other (BP_2), resulting in a complex SV with three breakpoints.

SVs can also be grouped in two categories depending on the resulting change in genome content. SV types such as inversions and translocations do not induce a change in genomic content, they are called *balanced* SVs. On the other hand, deletions, insertions, duplications and repetitions either remove or add genomic content, they are thus called *unbalanced* SVs. The concept of reference genome to define SV types is crucial for unbalanced SVs, where it allows to differentiate deletions from insertions.

2.4 Impacts of structural variants

While SVs are the least numerous category of variants in genomes - compared to the more frequent single-nucleotide polymorphisms (SNPs) -, their size that can go up to several Mbp makes them the more represented variants in terms of affected base pairs in most genomes. Their impact depends on both their type and location. Unbalanced SVs can modify the total genome size, a great example of which being bursts of retrotransposons activity (that occur as copy-and-paste events) that have been reported in several genus as a response to environmental stress (Belyayev, 2014). Regardless of genome size change, the main driver to study SVs is their

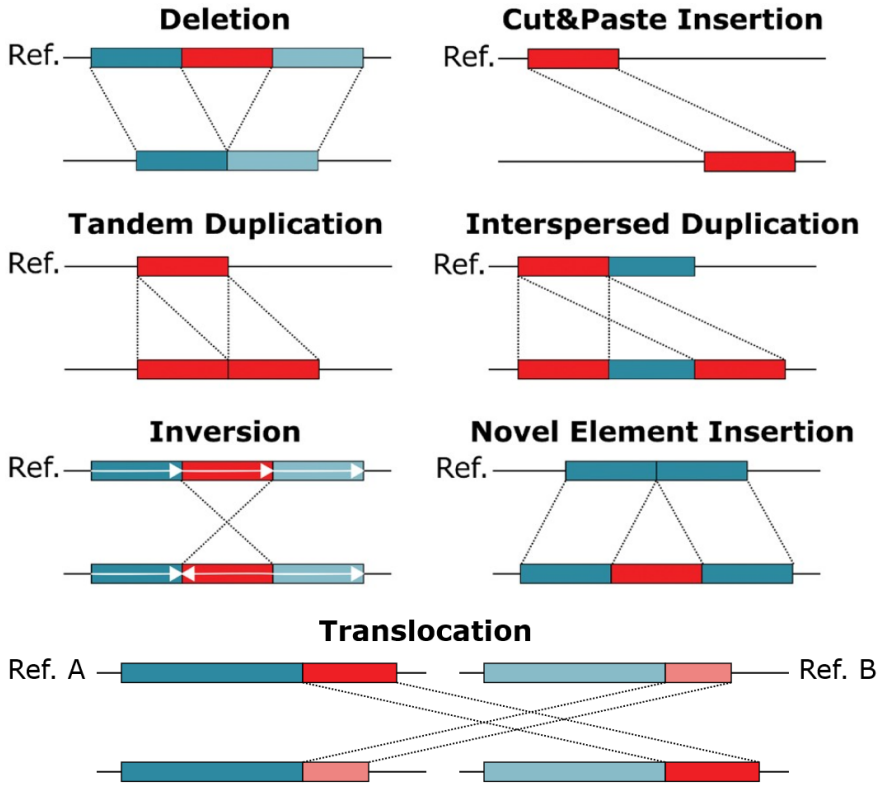

Figure 1 – Basic types of structural variants. The two rows for each SV type represent the alleles of the SV, with the upper one being the reference allele ('Ref.'). Red boxes correspond to the genome portion between the SV breakpoints. Adapted from Heller & Vingron (2019) Heller and Vingron (2019).

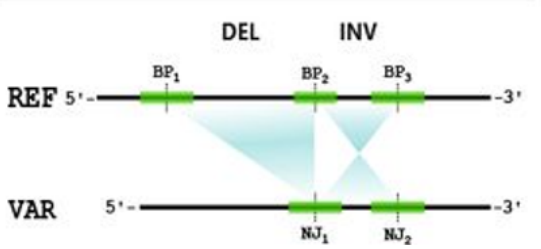

Figure 2 – Example of a complex SV. The deletion event shares a breakpoint with the inversion event, producing a compound SV with three breakpoints. Illustration borrowed from Mirus et al. (2024).

potential to affect the *phenotype*[2] when located around genes or gene regulation *loci*

[2]The phenotype can be viewed as a combination of 'outer' characteristics, by opposition to the genotype which describes the 'inner' genomic information. The phenotype of a cell can

(*i.e.* genomic localization). Genes are loci that carry the genetic information that has the potential to be *expressed* (*i.e.* translated into proteins, whose functions play a role in the phenotype), and gene regulation loci control gene expression (*e.g.* promoting expression, repressing expression). SVs that are located around gene loci can result in the apparition of novel genes and gene functions (through gene fusion or gene duplication) but can also result in gene loss. SVs located in the vicinity of gene regulatory loci can on the other hand modify gene expression.

3 Methodological problems for variant analysis

Characterizing variants is done as two successive steps: the variants are first discovered from one or a few sample genomes (*i.e.* *variant calling*), and then quantified in populations (*i.e.* *variant genotyping*). These two tasks call on bioinformatic problems such as whole-genome comparison and read mapping, which are heavily impacted by the characteristics of the genomic sequences available (*i.e.* sequence size and error rate).

3.1 Retrieving the sequence of a genome

3.1.1 Genome sequencing

In order to compare any set of genomes, one first need to get their sequence. The first generation of DNA sequencing technology was Sanger sequencing Sanger et al. (1977). Since then, high-throughput technological methods were developed, first with second generation sequencing (also called Next-Generation Sequencing, NGS) in the 2000s, then with third-generation sequencing (2008-2009). However, they all show technical limitation on the maximum sequence length they can produce. Indeed, genome sequencing requires preliminary *fragmentation* of the DNA into smaller molecules, that are then converted into digital sequences called *reads*.

The characteristics of sequenced reads differ on three main aspects between the (still evolving) technologies, that are the read length, the sequencing errors (both error rate and type of errors), and the sequencing cost of the technology.

In practice, reads are mainly classified based on their length between *short reads* produced by second-generation sequencing, that do not exceed a few hundreds of bp, and *long reads* produced by third-generation sequencing, with an average size of tens of thousand bp that can reach 1 Mbp for ONT ultra-long reads. Short reads display a minimal sequencing error rate, that improved from 1% to under 0.1% for the

describe the cell type or functions. The phenotype of an organism can describe its morphological or behavioral characteristics.

widely used Illumina sequencing (with an average length of 150 bp), and have seen their cost considerably reduced over the years, making them the most affordable to produce. On the other hand, the earlier versions of long reads contained a higher error rate, from 10 to 30% with the ONT and PacBio technologies, and were more expensive to produce than short reads. This error rate was progressively decreased with improvement of the technology, down to less than 1% with the latest PacBio CCS (also known as PacBio HiFi) and the latest ONT flowcells, though the latest technology comes with a higher cost. Although long reads are more expensive and erroneous than short reads, they improve considerably the quality and completeness of genome assemblies (see following Section 3.1.2) as they span more widely over highly repetitive regions (tandem repeats, transposable elements), as well as SV discovery and genotyping.

It is commonly done in practice to sequence genomes with a high enough *sequencing depth* (*i.e.* the average number of times any random base is sequenced) to ensure that the occasional (and mostly random) sequencing errors are underrepresented compared to the real sequence per genomic position. It is calculated as the number of reads multiplied by the average read length and divided by the genome length. A sequencing depth of 10 is expressed as 10X. It is an average estimate as all regions of the genome are not guarantied to have the same sequencing *depth*.

3.1.2 Genome assembly

After sequencing and few data cleaning and read-corrections steps, the reads are *assembled* to produce a genome. Genome assembly takes advantage of the fact that the genome is not fragmented at the same location, thus producing reads whose sequence *overlap* each other. Assembling a genome is done by finding overlaps between read extremities and combining the overlapping sequences into larger *consensus* sequences, called *contigs*. Some genomic regions are more difficult to assemble, in particular highly repetitive regions (such as telomers and centromers, *i.e.* chromosome extremities and chromosome centers), resulting in gaps in the assembly, which explains why the number of contigs is usually higher than the number of chromosomes in the genome. The contigs can be further ordered and oriented to form larger sequences called *scaffolds* if complementary evidence supports it (*e.g.* additional sets of larger reads, long range data such as Hi-C or optical maps). A chromosome-level assembly is reached when the number of scaffolds equals the number of chromosomes of the target genome.

De novo assembly, which is the task of genome assembly without relying on a species reference genome as a guide, is the most reliable way to retain potential rearrangements in the assembled genome. *De novo* assembly has been made easier with long reads (Koren et al., 2017; Li and Durbin, 2024), which produce longer overlaps between reads, and help increasing the contiguity of assemblies,

in particular in repetitive genomic regions. A near-complete assembly can be produced with long reads data, with a small number of gaps located in the most challenging regions of the genome. For a heterozygous diploid genome, a so-called primary haploid sequence is produced as the best representative consensus of the two haploid sequences, but it may include switches between haplotypes and miss genomic variants not carried by both of them. Nowadays, recent tools with high quality long-reads data alone or complemented with parental data or Hi-C can produce dual assembly pair, which aims to represent a pair of haploid genomes, thus including all genomic variations and often used for pangenome construction (see section 3.3.3) (Cheng et al., 2021; Li and Durbin, 2024).

3.2 Comparing genomes

Genomic sequences are traditionally compared using *sequence alignment*, either between genome assemblies (*i.e.* whole-genome alignment), or between a set of sequenced reads and a reference genome (*i.e.* read mapping). In the result of a *pairwise* sequence alignment, the two sequences are put on top of each other. Each nucleotide in both sequences is attributed a position in the alignment, while retaining the nucleotide order that is given by the input sequences. The alignment allows to identify where the sequences are identical (*i.e.* nucleotide matches), where they vary through nucleotidic polymorphism (*i.e.* nucleotide mismatches), and where they were offset by deletions or insertions (*i.e.* gaps). The number of matches, mismatches and gaps is used to compute the *score* of the alignment. The alignment score is obtained by attributing weights to matches (usually positive), as well as to mismatches and gaps (usually negative) and calculating the weighted sum.

There can be a multitude of possible alignments between two sequences, the problem of sequence alignment is to find an *optimal alignment* that maximises the alignment score. This problem is discussed in the context of whole-genome alignment in Section 3.2.1. An additional problem for read mapping is to first find the position of the read on the genome before performing the alignment, which is discussed in Section 3.2.2. Lastly, we touch on how alignment is used for variant calling, and the particular difficulty of structural variant calling compared to small variant calling in Section 3.2.3.

3.2.1 From assemblies: whole-genome alignment

There are two alignment approaches, either trying to find an optimal alignment covering the entirety of the input sequences - *global alignment* without rearrangements -, or trying to find optimal alignments of portions of the input sequences -

local alignment -. Two well-known algorithms providing the exact solution cover these two approaches using dynamic programming: Needleman-Wunsch algorithm Needleman and Wunsch (1970) for global alignment, and Smith-Waterman algorithm Smith and Waterman (1981) for local alignment. As rearrangements occur between genomes, the collinearity is not respected and so genome comparison through global alignment is not possible. WGA tools thus use the local alignment strategy (Armstrong et al., 2019). The WGA problem has been a major drive in the development of new strategies and tools over the years.

The first encountered limit for WGA has been its computing time. The exact solution Smith-Waterman algorithm is too time consuming to be used with whole genome sequences. Strategies using the principle of '*seed-and-extend*' have been developed in order to fasten the alignment speed. The *seed-and-extend* heuristic avoids the computation of all possible alignments by first identifying *seeds* (*i.e.* common words between the compared sequences). The seeds act as alignment *anchors*, they produce a reduced number of alignment initialization. The seed alignments are then extended on both of their extremities along the two sequences, stopping when the alignment score reaches a fixed minimal value. This heuristic has notably been used by BLAST, to fasten the search of sequence similarities among a database (Altschul et al., 1990), and was re-employed by many genome-to-genome and read-to-genome alignment tools such as MUMmer (Marçais et al., 2018), BLAT (Kent, 2002) and Last (Kiełbasa et al., 2011).

However, the main problem of WGA is the unavoidable trade-off between sensitivity (*i.e.* finding all the orthologous local alignments) and specificity (*i.e.* avoiding spurious local alignments resulting mostly from repeats). A strategy introduced to improve the specificity of WGA was *chaining*. The chaining of locally collinear local alignments into larger synteny blocks allows to filter out spurious alignments, while identifying homologous and rearrangement-free regions as well as rearranged regions (Darling et al., 2004; Peng et al., 2009; Marçais et al., 2018; Minkin and Medvedev, 2020).

3.2.2 From reads: read mapping

Because *de novo* genome assembly has been an arduous problem up until the arrival of the latest generations of long-read sequencing technologies, genome comparison has mostly been done by *read mapping* on a reference genome. The problem of read mapping is two-fold: (1) each read must first be localized on the reference genome, before being able to (2) identify the differences between the read sequence and the reference genome sequence. Once again, seed-and-extend strategy was adopted for the mapping of both short and long reads. However, as long reads contain more sequencing errors than short reads, the size of the seeds used needs to be smaller, which increases the number of spurious mappings caused by repeats. The main

strategy used to try and overcome this limit is *sketching* (Sahlin et al., 2023). In the context of read mapping, sketching compresses the information contained in a read's *k-mer* set into fingerprints representative of the read's sequence. Several sketching techniques have been developed, among which *minimizers*, which are used by the popular sequence aligner and read mapper minimap2 (Li, 2018). Given a window size w and a k-mer length k, a set of w consecutive k-mers is represented by the k-mer with the minimal rank following a given ordering function (*e.g.* lexicographical order), which becomes the minimizer of this window. Minimizers and other sketching techniques have also the advantage of reducing the number of anchors and thus accelerate the reads alignments. On top of this different seeding strategy between short and long read mapping, long read mapping also comprises a chaining step on the alignment anchors (Figure 3).

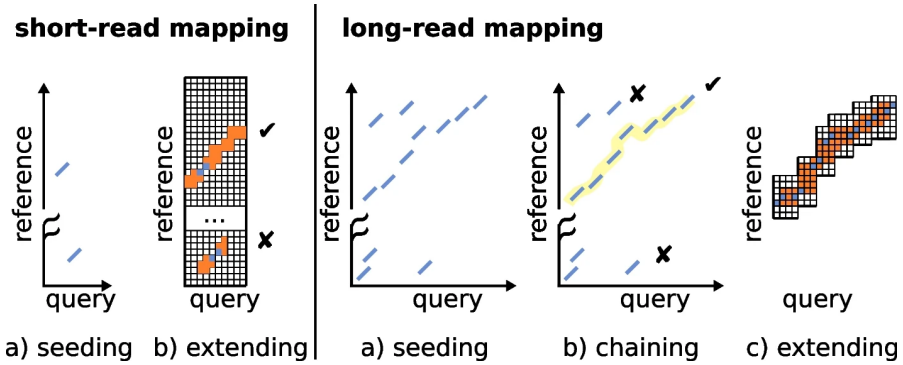

Figure 3 – Illustration of the short and long read mapping steps. From Sahlin et al. (2023).

3.2.3 Variant calling and genotyping

Variant calling is traditionally done by aligning reads on a reference genome. Small variants can then be discovered from mismatches or gaps within the resulting alignments. For instance, single nucleotide polymorphisms (SNPs) can be called from punctual mismatches at specific locations on the reference genome that are shared between multiple reads. However, detecting structural variants is more challenging because their size causes the reads to align in multiple segments on the reference genome. It is the gaps and difference of alignment orientation between these blocks that can indicate the presence and type of SVs, rather that the base alignments of the reads. The specific SV calling methods are detailed in Section 1 of Chapter II.

Variant genotyping is the task of predicting the alleles carried by one or several individuals for a given set of known variants. In a diploid genome, SVs have three

possible genotypes. They can be present as two copies of the reference allele (*i.e.* reference *homozygous*), two copies of the alternative allele (*i.e.* alternative *homozygous*), or one copy of each allele (*i.e. heterozygous*). The end goal of population-wide genotyping (*i.e.* genotyping variants in a large number of individuals from a given population) is to discover associations between variants and phenotypes. These analyses are typically done using matrices representing the variants in lines and the individuals in columns (or vice versa), and containing in each cell the genotype of the given variant in the given individual. In this thesis, we will focus on SV genotyping methods based on read mapping, that are presented in Section 2 of Chapter II.

3.3 Representing the genomic variability

3.3.1 VCF and reference genome

For a long time, variants have been represented in a coordinate referential based on a reference genome. The VCF format was dedicated to store all types of variants, from SNPs to SVs (Danecek et al., 2011), and is the most prevalent format used to this end. It is a tabulated format, reporting one variant per line, with each column containing a designated information (*e.g.* reference chromosome, nucleotidic position of the starting breakpoint, sequence of each allele). Its strength lies in its wide versatility, as custom annotations can be reported using any custom tag, as long as those tags are defined in the header of the file. Notably, the VCF format can also store the genotype predictions of variants in sample genomes. An illustration of the VCF format is presented in Figure 4.

While VCFs allow for an extensive description of variants (*e.g.* allelic frequencies of SNPs, imprecision of SVs breakpoints), they condition the variant information to be reported relatively to a single reference genome. Furthermore, even with available knowledge on the genomic variability in a species, read mapping is still constrained to be performed on a linear reference sequence. This approach is biased, as it makes it difficult to identify variants that are absent from the reference but shared by other individuals. This reference bias can limit the ability to discover new variants in populations mildly distant to the reference genome used, and biases the genotype inference towards the reference genotype.

3.3.2 Variation graphs

Variation graphs, as proposed by Garrison et al. (2018), were introduced to overcome reference bias by representing the genome and its variants in a graph structure, which can be used for mapping reads. Variation graphs are directional sequence

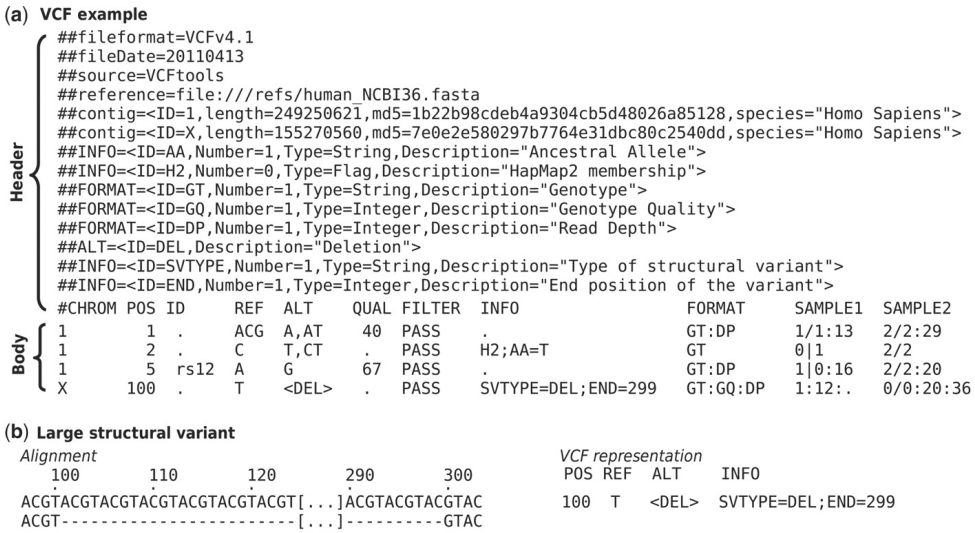

Figure 4 – Illustration of the VCF format. (a) General organization of a VCF file. The header lines contain the metadata of the file (*e.g.* reference genome, description of the chromosomes) and the definition of the tags used in the body lines. The body lines describe the variant entries, one line par variant. (b) Illustration of a deletion and its representation in the VCF. SV entries use specific tags in the INFO field (here "SVTYPE" and "END") which are not needed for SNPs or INDELs. Adapted from Danecek et al. (2011).

graphs, their nodes are labeled with sequences, and their edges represent the sequence adjacencies in genomes. They are built from a reference genome, whose sequence is represented in its entirety in the graph, and a set of known variants, represented as *bubbles* in the graph. The individual sequences can be reconstructed by following the path in the graph including each individual node from its source node to its sink node. At a variant site, a new walk is added, from the node preceding the variant start position, to the node following the variant end position, forming a bubble (Figure 5). The GFA format has been proposed as a standard to store variation graphs to use among variation graph-based tools[3].

The purpose of variation graphs is to improve read mapping, by representing the genome sequence and its variability in a single data structure that can be used as a base for mapping. Several mapping tools on such sequence graphs have consequently been developed: Giraffe (Sirén et al., 2021) for short reads, GraphAligner (Rautiainen and Marschall, 2020) and minigraph (Li et al., 2020) for long reads. The results of sequence alignment on graphs are commonly stored in

[3] https://github.com/GFA-spec/GFA-spec

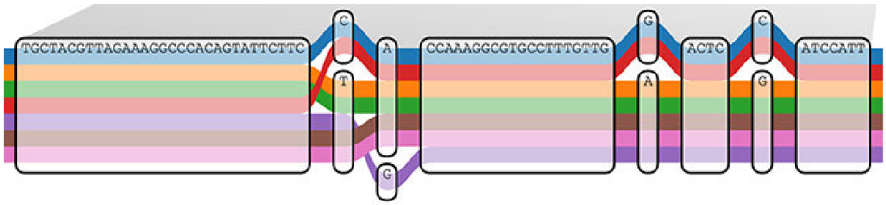

Figure 5 – Illustration of a subset of variation graph. Adapted from Garrison et al. (2018).

the GAF format[4], derived from the PAF format, which is one of the formats used for sequence alignment on linear reference.

3.3.3 Pangenome graphs

Pangenome graphs are the latest way of representing the variability in genomes (Li et al., 2020; Hickey et al., 2024; Garrison et al., 2023). Their graph structure is essentially the same as that of variation graphs, and they are also stored using the GFA format. However, they are constructed from WGA over a collection of entire genomes. As such, they can be constructed without prior knowledge on the variants that are to be represented in the graph. This means that the completeness and accuracy of the variants represented in the graph are not dependant on that of SV discovery tools, and they can represent variants larger that what can be detected from reads.

4 The special case of inversions

4.1 Inversions impacts and dynamics in genomes

Inversions are a type of SV characterized by the change of orientation of a chromosomal fragment. They have been particularly studied in flies, where 8 large inversions (of average size of 8.9 megabases) cover 43% of the *Drosophila melanogaster* genome (Kirkpatrick, 2010). As a whole, inversions appear to be generally larger in size than other types of SVs, with a size that can reach several megabases (Mb) in plants and animals (Wellenreuther and Bernatchez, 2018). Furthermore, inversions can impact genomes and populations in more ways than the more common SV types of deletions and insertions. When inversions are in the heterozygous state in a diploid genome (i.e. presence of both the ancestral and inverted alleles), a single

[4] https://github.com/lh3/gfatools/blob/master/doc/rGFA.md#the-graph-alignment-format-gaf

crossover during meiosis within the inversion generates unbalanced gametes that contain duplication and/or deletion. As a result, the effective rate of recombination in such region is highly reduced. Thus, the spread or fixation of such underdominant large inversions is likely driven by natural selection, suggesting that they include genes related to adaptative traits (Kirkpatrick, 2010; Huang and Rieseberg, 2020; Berdan et al., 2023). This local impairment of recombination allows them to 'capture' and maintain new combinations of genes and loci. Large inversions in particular, can capture more genes and so may have a higher probability of capturing non neutral allelic combinations than small inversions (Wellenreuther and Bernatchez, 2018).

At the evolutionary level, this reduced recombination rate between the two alleles of an inversion can promote adaptation by maintaining advantageous allele combinations, or even cause an accumulation of mutations leading to reproductive isolation (Berdan et al., 2023). An extreme example of phenotypic polymorphism related to inversions capturing specific allele combinations is supergenes, which capture multiple phenotypic characters while maintaining balanced polymorphism in the populations (Thompson and Jiggins, 2014). A great example of supergenes is the P supergene of the *Heliconius numata* butterfly (Figure 6), which contains two large inversions and is involved in locally-adaptated wing pattern polymorphism (Joron et al., 2011).

Inversions, among genomic variants, have an exceptional potential to drive the spread of advantageous or adaptive alleles and traits, introduce and maintain phenotypic polymorphism, or even contribute to speciation in populations and species. Still, it is important to note that most of the inversions rearrangement events that occur in genomes are likely to have a deleterious impact and to be counter-selected. It is those inversions that remain in genomes - those than can be observed - that might influence the evolution of species, either under neutral or advantageous selection.

4.2 Introduction of the alpine *Coenonympha* butterfly species

The alpine *Coenonympha* butterflies form a complex of four species spread along a geographical and altitudinal gradient in the mountain chain of the Alps (Capblancq et al., 2019). *C. arcania* (Pearly Heath) is most most widespread species of the complex, found from sea level to elevations around 1,500 m and not restricted to the Alps, while *C. gardetta* (Alpine Heath) typically occupies higher elevations of more than 1,500 m in the Alps, the French Massif Central and the Balkans. The two other species of the complex, *C. darwiniana* and *C. cephalidarwiniana* are found at similarly high elevations to that of *C. gardetta*, but each of them occupy a distinct geographic area where they replace *C. gardetta* (Figure 7).

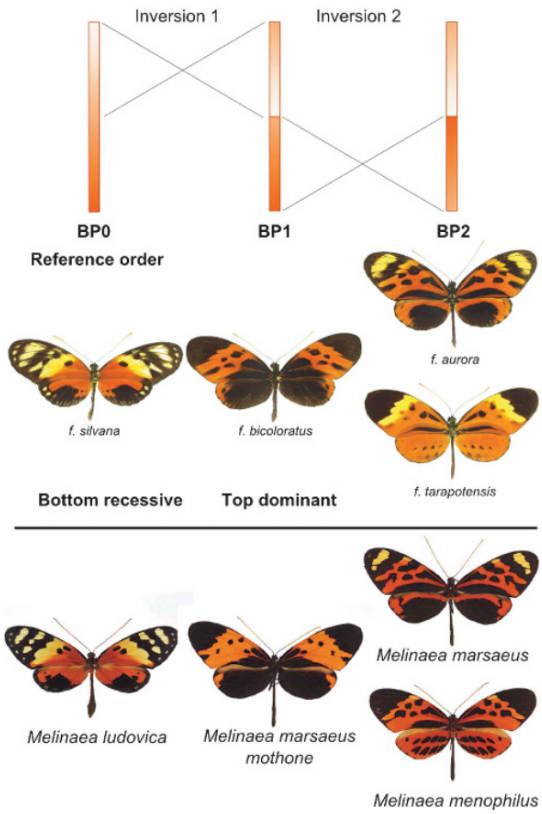

Figure 6 – Example of the *Heliconius numata* P supergene, and its associated wing pattern polymorphism. The three combinations of alleles for the two inversions (BP0, BP1, BP2) are each associated with specific wing patterns in the butterflies. Adapted from Thompson and Jiggins (2014).

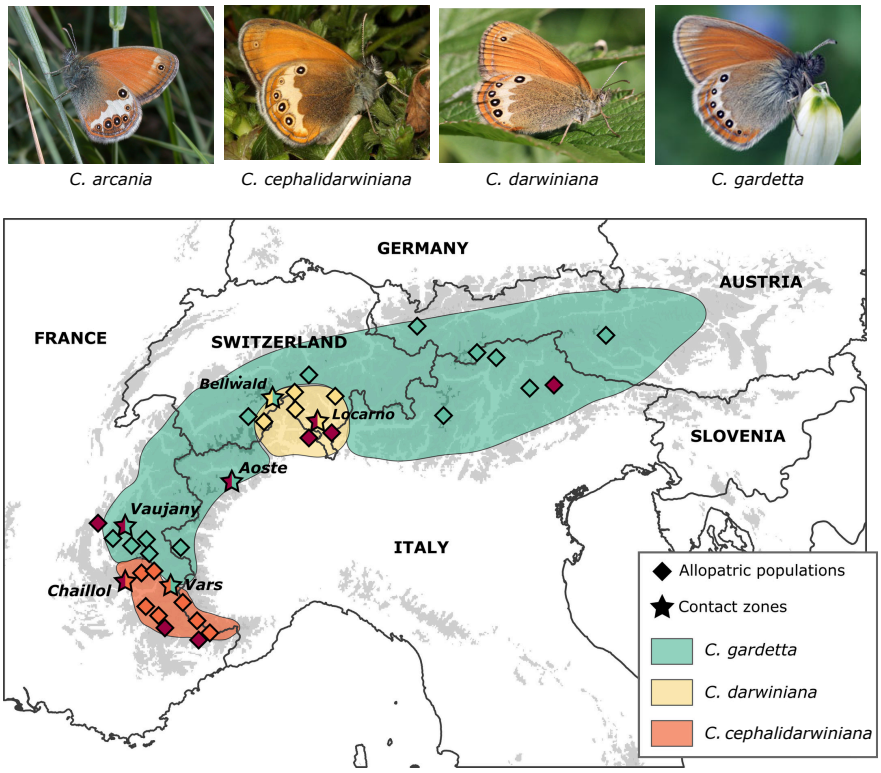

Figure 7 – Morphology of the four alpine *Coenonympha* species and distribution range of *C. gardetta*, *C. cephalidarwiniana*, and *C. darwiniana*. The range of the fourth species, *C. arcania* (everywhere in Europe at low elevation, including the valleys of the Alps) is not shown. Adapted from Capblancq et al. (2019). Photography credits: Daniel Morel (*C. arcania*), Claire Hoddé (*C. gardetta*), Wolfgang Wagner (*C. cephalidarwiniana*) and Matt Rowlings (*C. darwiniana*).

These four species are closely related. While the divergence between *C. arcania* and *C. gardetta* is approximated to have happened around 1.5 to 4 million years ago, *C. darwiniana* and *C. cephalidarwiniana* were found to most probably be the product of hybridization between *C. arcania* and *C. gardetta*, that took place around 10,000 to 20,000 years ago (Capblancq et al., 2015). Analysis of SNP data identified the most probable scenario as that of an ancestral hybrid population that underwent genetic differentiation into two hybrid lineages, currently in an advanced stage of the hybrid speciation process (Capblancq et al., 2015, 2019). Although they present strong genetic isolation with the parental species *C. arcania*, the two hybrid species were observed to be interfertile with *C. gardetta* in contact zones (*i.e.* where their geographical range overlap, shown in Figure 7) (Capblancq et al., 2019).

The genomic factors contributing to the speciation between the hybrid lineages or to their adaptation to climatic and altitudinal conditions more similar to one of the two parental species are still little-known. Analysing the genomic rearrangements between these species could help bring new insights. However, past studies (Capblancq et al., 2015, 2019, 2020; Després et al., 2019; Sherpa et al., 2022; Kebaïli et al., 2023) mainly employed ddRAD-seq Illumina data and no genome has yet been assembled for any of these species. The closest available genomes are that of *Coenonympha glyceryon* (ENA project PRJEB71111), *Maniola jurtina* and *Pararge aegeria* from the Darwin Tree of Life project (Lohse et al., 2021b,a).

5 Thesis objectives

This thesis is a research work on methods to discover, genotype and represent SVs, and was guided by specific data (of the alpine *Coenonympha* butterflies) and biological application. More specifically, the thesis objectives were to develop novel methods to detect and genotype SVs making use of the new data structures that are variation and pangenome graphs. This thesis is part of the DIVALPS project, supported by the *Agence Nationale de la Recherche* (ANR), which takes interest in the evolution history of the four alpine *Coenonympha* species and in the genetic and genomic factors involved in this evolution. This biological model focused my work on whole-genome comparison, in a context of non model, inter-specific genomes.

II State of the Art

The aim of this chapter is to present the methods that have been developed to this day to characterize structural variants in populations. This characterization can be divided into two distinct problems, namely variant *discovery* and variant *genotyping*, to each has been dedicated a section of this chapter. The third section of this chapter focuses on *pangenome graphs*, the currently available methods to construct them and how they can be used to characterize structural variants.

In this chapter

1 Structural variant discovery 20
- 1.1 From sequencing reads to genome assemblies . . . 20
- 1.2 Methods for SV discovery from *de novo* assemblies . . . 22
  - 1.2.1 Whole-genome alignment . . . 23
  - 1.2.2 Assembly-based SV discovery tools . . . 24

2 Structural variant genotyping with long reads 26
- 2.1 General strategy for structural variant genotyping . . . 26
- 2.2 Two categories of long-read genotyping tools . . . 27
  - 2.2.1 'Force-calling' genotyping with discovery tools . . . 27
  - 2.2.2 Genotyping-dedicated tools . . . 28
- 2.3 Current limits of structural variant genotyping . . . 29

3 Pangenome graphs: a new era for the characterization of SVs 30
- 3.1 Advantages of pangenome graphs over variation graphs . . . 31
- 3.2 Pangenome graph construction . . . 31
- 3.3 Structural variants in pangenome graphs . . . 32
- 3.4 Current limits of SV characterization in pangenome graphs . . . 33

1 Structural variant discovery

1.1 From sequencing reads to genome assemblies

Identifying structural variants in Whole Genome Sequencing (WGS) samples involves aligning the sample's genomic sequence to a reference genome. This alignment can be performed either directly using the raw reads (*e.g.* short or long reads) or after assembling the reads into contigs. Before the era of PacBio High Fidelity (HiFi) sequencing, SV discovery was mainly done by read mapping, as *de novo* genome assembly was challenging due to the small length of short reads (*i.e.* Illumina) or the high error rate of long reads, and either way required significant computational resources.

A key factor in SV discovery is the length of the sequence fragments used. SVs are more likely to be detected if the sample's sequence spans their entire length, and longer sequences can map with higher accuracy on repetitive regions of the genome. Repetitive sequences (repeats), which are for the most part SVs, occupy a large portion of many eukaryote genomes (e.g. 50% of the human genome) and are a major obstacle in SV discovery, particularly with short reads whose length is smaller than most genomic repeats. Moreover, between 80% and 90% of the deletions and the insertions identified by long reads in the human genome are localized in regions with a repetitive context (Zhao et al., 2021; Chaisson et al., 2019; Delage et al., 2020). Consequently, the improvement in SV discovery brought by the introduction of long read sequencing technologies has been unanimously observed in the literature (Mahmoud et al., 2019; Zook et al., 2020; Zhao et al., 2021; Audano et al., 2019; Delage et al., 2020), both through an increase of the discovery rate (*recall*) and a decrease of the false positive rate (*i.e.* increase of *precision*). In their review of SV discovery methods, Mahmoud et al. (2019) show that long reads increased the number of discovered SVs by at least two-fold compared to short reads in several genome models (Fig. 8). Improvement of SV discovery recall with long reads over short reads is also particularly marked for insertions, where short read data was able to detect less than a third of the insertions called by long reads and generally do not resolve their complete sequences (Delage et al., 2020).

The most popular (though not exhaustive) tools for SV discovery based on long read mapping are PBSV [1], Sniffles (Sedlazeck et al., 2018; Smolka et al., 2024), SVIM (Heller and Vingron, 2019), and cuteSV (Jiang et al., 2022). The general method of SV discovery with long (and short) reads consists in finding *SV signatures* in the mapping of the reads on the reference genome. SV signatures are heavily based on *split reads*, where two (or more) consecutive portions of a read map disjointedly on the reference. The distance and orientation between these

[1] https://github.com/PacificBiosciences/pbsv

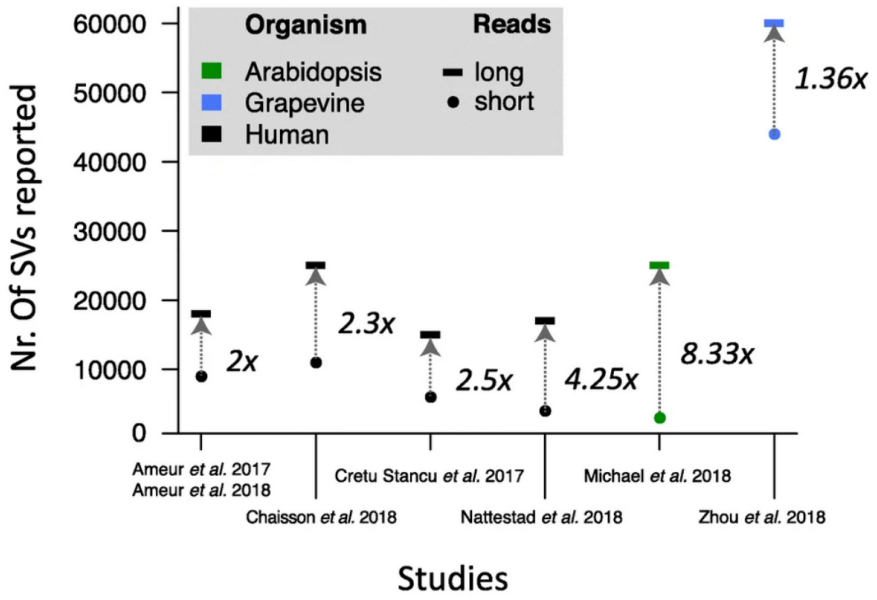

Figure 8 – SV discovery performances of short and long read sequencing. This figure shows the increase of discovered number of SVs from short to long read calling observed in different studies. Adapted from Mahmoud et al. (2019)

disjoint mappings can differentiate between multiple SV types, as illustrated in Fig. 9. For instance, a change of orientation across split mappings is a signature for an inversion event, while a gap between split mappings is a signature for a deletion event. Usually, more than one read provide mapping signature for a single SV, and sequencing errors can produce SV signatures with erroneous positions. This generates redundancy in the set of signatures leading to imprecise SV calls. In order to correct this, all SV-caller using long-reads have a clustering step which merges the signatures based on the type, position, and length of the corresponding SVs. Interestingly, the read mapper impacts the recall and precision of the SV discovery with long reads. Liu et al. (2024) demonstrated that the best results are obtained with specific mapping-discovery pipelines, and that these preferential pipelines differ depending on the SV type. For instance, the long-read mappers winnowmap (Jain et al., 2022) and minimap2 (Li, 2018) combined with cuteSV or Sniffles produce the best results for the discovery of deletions, while NGMLR (Sedlazeck et al., 2018) and pbmm2[2] combined with SVIM or SVision (Lin et al., 2022) produce the best results for the discovery of inversions.

More recently, alternative approaches based on deep-learning or alignment-free were developed. For instance, SVision (Lin et al., 2022) converts the read mappings

[2] https://github.com/PacificBiosciences/pbmm2

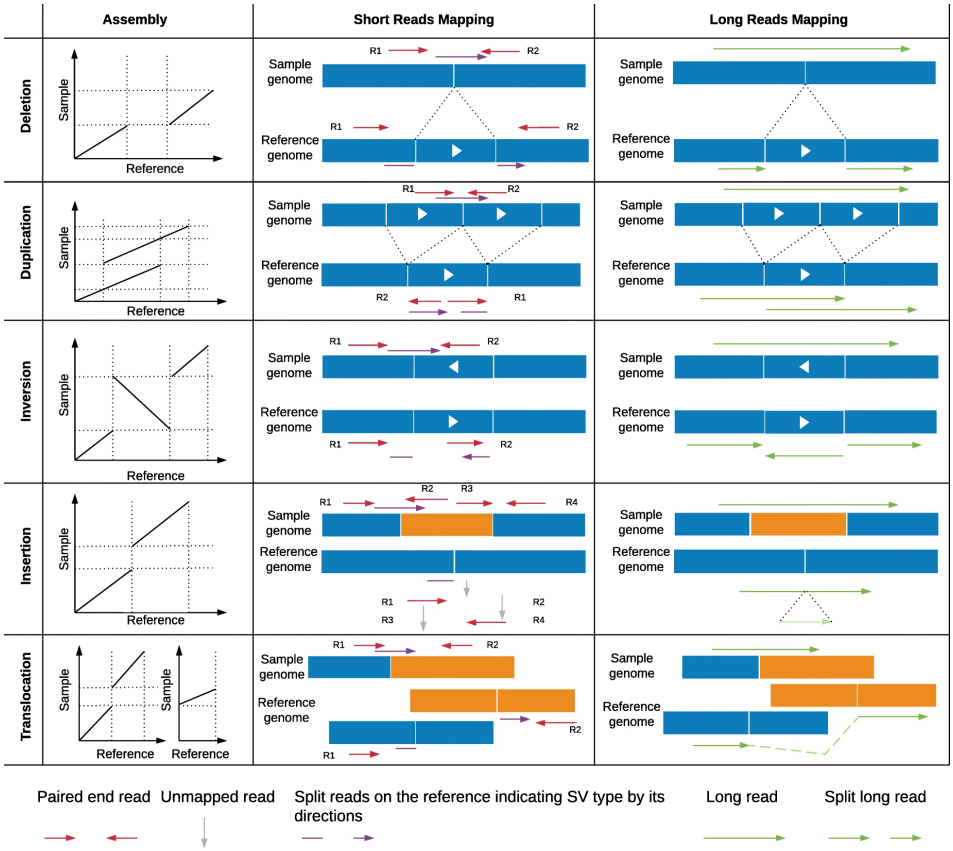

Figure 9 – SV signatures used for SV discovery with short and long reads. From Mahmoud et al. (2019).

on the reference genome into graphical representation, that can be analyzed by a deep-learning architecture trained to recognized the different variant types from the mapping pattern of the reads. Because the alignment-based methods are mostly developed and tested on human data, and are highly dependant on the quality of the read alignments, alignment-free approach such as SVDSS (Denti et al., 2023) have been developed. SVDSS improves SV discovery in repetitive regions of genomes In short, SVDSS identifies sample-specific strings in two sets of reads to infer the unique sequence adjacencies denoting SV breakpoints.

1.2 Methods for SV discovery from *de novo* assemblies

While long reads indeed brought an improvement for SV discovery, some SVs still exceed the size of long reads, and large insertions and complex SVs can still be challenging to discover. Thanks to the recent decrease of long read sequencing error

rates (*e.g.* PacBio CCS) and the development of new long-read assembly tools with decreased runtime (Koren et al. (2017); Cheng et al. (2021)) has rendered *de novo* assembly more accessible. With the achievment of high quality reads (*i.e.* HiFi reads), ultra-long and Hi-C reads, and the development of new software, *de-novo* chromosome-level assembly is more accessible with decreased runtime (Koren et al., 2017; Cheng et al., 2021; Li and Durbin, 2024). Whole genome comparison can thus now be used more routinely for SV discovery, the assembly contigs acting as even longer reads with less erroneous bases. The higher base-level accuracy of assemblies compared to long reads allows for more precise inference of SV breakpoints, and their greater length improves the discovery of large-scale rearrangements in genomes and does not require a clustering step (Ahsan et al., 2023). However, haplotype-resolved (phased) assemblies can still be challenging to produce for highly heterozygous or polyploid genomes. This poses a problem for the discovery of heterozygous SVs, as for each variant, the allele included in an unphased (haploid) assemblies is arbitrary. Overall, half of the heterozygous variants are lost when called on unphased assemblies of diploid genomes.

1.2.1 Whole-genome alignment

Discovering SVs from assembly alignment introduces the problems of whole-genome alignment (WGA) described in section 3.2.1. At this day, there are many WGA tools available, catering for different applications (*e.g.* intra-species comparison with minimap2 (Li, 2018), nucmer from the MUMmer suite (Marçais et al., 2018) or GSAlign (Lin and Hsu, 2020); inter-species comparison with ProgressiveCactus (Armstrong et al., 2020) or AnchorWave (Song et al., 2022)). Historically, WGA tools were designed for the comparison of highly divergent, inter-specific genomes. In such context, identifying precise breakpoints and/or SVs was not possible or needed. As such, these inter-specific WGA tools, as well as the more recent intra-specific WGA tools, output pairwise alignments or synteny blocks, but not a list of characterized SVs.

That is why a series of tools have been developed to characterize variants from WGA, described in section 1.2.2. These tools advise the use of two specific sequence aligners, Minimap2 and nucmer. Both of these tools perform local alignment and collinear chaining. Minimap2 extracts the minimizers of the input sequences and uses them to find anchors (*i.e.* exact matches) on the reference sequence for each query sequence. Collinear anchors are combined into chains and gaps between adjacent anchors are closed by alignment extension through dynamic programming. Minimap2 has different presets of parameters adapted to different query types (*e.g.* long reads, assembly) and to different nucleotidic divergence levels (*e.g.* 0.05%, 0.1%), though it is primarily intended to be used for same-species alignment.

Although the transition from read-based to assembly-based SV discovery opened

the door to inter-species genome comparison, one of the current limits of assembly-based SV discovery is that these tools are all designed to work with aligners mainly suited for the intra-species scale.

SibeliaZ. SibeliaZ (Minkin and Medvedev, 2020) finds homologous sequences by constructing a de Bruijn graph from the input sequences and finding "carrying paths" in the graph, which are defined as paths that go through the most frequently visited vertices. Its interesting feature is the output of homology blocks and their order in the input genomes. These blocks can be chained into larger synteny blocks using its companion tool maf2synteny (Kolmogorov et al., 2018). Although there is currently no automatized method to analyze these synteny blocks to discover SVs, it is a promising avenue to discover large scale rearrangements (*e.g.* large inversions) between genomes whose divergence is higher than what is handled by minimap2 or nucmer.

1.2.2 Assembly-based SV discovery tools

Assembly-based SV discovery tools employ a similar approach as that from long reads-based discovery tools. Alignments of the assembled contigs of the sample against the reference genome are analyzed in search of variant *signatures*. For instance, gaps in the alignment can reflect the presence of deletions or insertions, split alignments with a change of alignment orientation can reflect inversions, and scattered alignments of a contig against two reference chromosomes can reflect translocations. The majority of state of the art tools (Assemblytics (Nattestad and Schatz, 2016), SyRI (Goel et al., 2019) and SVIM-asm (Heller and Vingron, 2020)) take WGA as input, requiring the genomes to be pre-aligned using minimap2 (Li, 2018) or nucmer (Marçais et al., 2018). MUM&Co (O'donnell and Fischer, 2020) performs genome alignment internally, using nucmer, and thus directly takes the two genome sequences as input. Aside from their shared approach of signature search, state of the art methods each present particularities of their own.

Assemblytics. Assemblytics exclusively takes alignments output in the nucmer-specific format. Its method starts by filtering the input alignments, only keeping alignments with at least a minimum amount of unique contig sequence anchor (10 kilobase pairs by default) that are found in no other alignment of the considered contig. Then, each pair of consecutive alignment along a sample contig is analyzed, the spacing and orientation between the two alignments determining the presence of variants and their type. Assemblytics can discover deletions and insertions of 1 bp to 10 kbp, as well as tandem and repeat expansions (*i.e.* insertions in tandem or repeat regions) and contractions (*i.e.* deletions in tandem or repeat

regions). Assemblytics is available as a web application and outputs the discovered variants in a BED file format. It also provides plots of variant size distribution and interactive visualizations of the alignments.

MUM&Co. MUM&Co performs a reciprocal alignment of the input genomes using the nucmer function of nucmer. From the alignments, it associates query contigs to reference chromosomes by identifying a subset of the most accurate, non-overlapping alignments, and identifies putative SVs from alignment signatures. Then it filters false-positives using the reciprocal whole-genome alignment and an inferred global-like alignment using the 'global' filter of the MUMmer suite (Marçais et al., 2018). For example, both query-to-reference and reference-to-query global alignments should display a gap that matches the positions of a putative inversion. MUM&Co also comes with a unique feature to optionally annotate mobile or novel genetic elements among insertions and deletions. A minor drawback of the tool is that it outputs the SVs in a tool-specific tabulated file format, instead of the commonly used VCF file format.

SVIM-asm. SVIM-asm is a tool derived from the long-read based SV discovery tool SVIM (Heller and Vingron, 2019). It employs the same method of signature search among input alignments. It provides the advantageous feature of handling phased diploid assemblies in addition to haploid assemblies. When given alignments from a diploid sample assembly, it discovers SVs independently for each haplotype, and then clusters SV calls from both haplotypes to infer a genotype to the discovered SVs.

SyRI. SyRI performs variant discovery in two hierarchical steps. It first constructs a graph from all forward alignments between a pair of homologous chromosomes, and then tries to identity the longest subset of syntenic aligned regions between the two extremities of the chromosome. The identified syntenic regions in turn designate the remaining, non-syntenic, regions as structural rearrangements. The rearrangements can then be annotated as inversions, translocations or duplications based on their alignment signatures. Its second step is to identify the smaller SVs (deletions, insertions, CNVs), indels and SNPs in both the syntenic and rearranged regions. All rearrangements and associated smaller variants are then output in a VCF file. One of the originalities of the tool is that it annotates the breakpoint positions of variants in both the reference genome and the sample genome. The other is its hierarchical take on variant discovery (*i.e.* first identifying rearrangements, then identifying small variants inside these rearrangements), which the authors argue is beneficial for downstream analyses, as variants inside structural rearrangements do not follow the same inheritance dynamics as variants in regular

regions of the genome. The drawback of the method compared to others is its need for a chromosome-level contiguity quality for both compared genomes.

From a user point of view, the tools differ mainly on the types and sizes of SV they are able to discover (Tab. 1).

Table 1 – Comparison of state of the art assembly-based SV discovery tools. For each tool are indicated the types of SV handled, whether the tool also reports small variants, the maximum size of reported SVs, the mandatory or preferred aligner for the tool and the output format. INS: insertions, DEL: deletions, DUP: duplications, TANDEM DUP: tandem duplications, TANDEM: tandem expansions and contractions, REPEAT: repeat expansions and contractions, INV: inversions, BND: translocations, CNV: copy number variants.

Tool	SV types	Small variants	Max. size	Aligner
Assemblytics	INS, DEL, TANDEM, REPEAT	indels	10 kb	nucmer
SyRI	INS, DEL, DUP, INV, BND, CNV	SNPs, indels	none	nucmer or minimap2
MUM&Co	INS, DEL, TANDEM DUP, INV, BND	no	none	nucmer
SVIM-asm	INS, DEL, DUP, INV, BND	no	none	minimap2

2 Structural variant genotyping with long reads

SV genotyping involves analyzing a set of reads from an individual or a group of individuals. The goal is to determine the genotype at each variant position (*i.e.*, which allele is present in the sample) with a confidence level that depends on the depth of coverage. While short read data is still used for large-scale population SV genotyping due to its lower cost, the benefits of long read data for SV discovery presented in Section 1.1 also apply to SV genotyping as most genotyping methods are also based on read mapping.

Population-wide genotyping is more efficient when performed for a fixed preset of SVs across a collection of resequencing samples, without having to first scan blindly the whole genome for each sample. So, the popular discovery tools Sniffles and cuteSV adapted their genotyping module to provide a 'force-calling' mode (*i.e.* standalone genotyping mode), and various tools specifically dedicated to SV genotyping with long reads were developed: VaPoR (Zhao et al., 2017), LRcaller (Beyter et al., 2021), SVJedi (Lecompte et al., 2020).

2.1 General strategy for structural variant genotyping

The general strategy to genotype a set of SVs from long reads, common among state of the art tools, is to make count of the reads that support each described allele,

for each SV. To this end, the long reads are first mapped to a reference genome. Then, alignments of reads for in the vicinity of each SV are selected. The selected reads can then be partitioned between the reads supporting the reference allele and the reads supporting the alternative allele(s). From the read count in each partition, the methods (at the exception of the former version of Sniffles, Sedlazeck et al. (2018)) calculate the likelihood for each potential genotype and identify the genotype with the highest likelihood as the predicted result. The output is a VCF file containing the input SV description with added fields reporting the predicted genotype and additional scores and statistics pertaining to the genotyping quality and confidence. The accuracy of the predicted genotypes relies on the accurate count of allele supporting reads. An accurate count implies an equal chance of the read mappings to reflect both the reference and alternative alleles, which creates an issue when the reads are aligned to a linear genome reference, meaning only the reference allele. State of the art methods primarily differ by their read selection and partitioning strategy. Table 2 synthesizes the common and differing characteristics and approaches of the long-read based SV genotyping tools, which are further detailed in Section 2.2.

Table 2 – Overview of the main method differences between long-read mapping-based SV genotyping tools. The "read selection" and "read count" columns denote the allele sequence to which the reads are compared at each of the two steps. "Ref only" means that reads are selected/counted based on their mapping to the reference allele only. "Ref + alt" means that the reads are selected/counted based on their mapping/comparison to both the reference and alternative alleles.

Tool	Genotyping mode	Read selection	Read count	Ref.
Sniffles	force-calling	ref only	ref only	Sedlazeck et al. (2018); Smolka et al. (2024)
cuteSV	force-calling	ref only	ref only	Jiang et al. (2022)
VaPoR	dedicated	ref only	ref + alt	Zhao et al. (2017)
SVJedi	dedicated	ref + alt	ref + alt	Lecompte et al. (2020)
LRcaller	dedicated	ref only	ref + alt	Beyter et al. (2021)

2.2 Two categories of long-read genotyping tools

2.2.1 'Force-calling' genotyping with discovery tools

The *force-calling* approach is proposed by discovery tools - Sniffles (Sedlazeck et al., 2018; Smolka et al., 2024) and cuteSV (Jiang et al., 2022) - to genotype SVs from a fixed set given as input. The screening for variant signatures, part of the discovery

(*calling*) step, is performed at genomic locations targeted by the input SVs position rather than on the whole genome. The tools require preliminary mapping and alignment of the reads against the reference genome, provided as a BAM formatted file. The reads that map on SV positions on the reference genome are selected, and their alignment to the reference are analyzed to identify SV signatures (*e.g.* from split read alignments). The force-calling method is not further described in the corresponding papers or documentation. We assume that it reuses the method of their regular discovery-genotyping mode, where reads are clustered based on their signature characteristics (*i.e.* positions of the signature, putative type of SV). Reads bearing a given SV signature are then counted as reads supporting the alternative allele of this SV, while reads with no signature (*i.e.* reads that match the reference at the SV location) are counted as reads supporting the reference allele.

2.2.2 Genotyping-dedicated tools

There are three tools dedicated to SV genotyping with long reads: VaPoR (Zhao et al., 2017), LRcaller (Beyter et al., 2021) and SVJedi (Lecompte et al., 2020). VaPoR and LRcaller, as Sniffles and cuteSV, take a BAM file of the alignment of the reads to the reference genome and a set of described SVs as input. However, instead of partitioning the reads based on their alignment to the reference only, they represent both reference and alternative alleles for each SV of the input set, and compare the sequence of the selected reads to the sequence of the represented alleles.

VaPoR. VaPoR compares the sequence of each selected read to both the reference and alternative alleles using an alignment-free method. It uses fixed-sized words (by default 10 bp) to construct one dot-plot per allele (*i.e.* matrices with one axis representing the position on the read, and the other axis representing the position on the reference or alternative allele), where a dot represents an identical k-mer in both sequences. Then, for each allele, a similarity score is calculated by summing the distances of identical k-mers to the dotplot diagonal (which represents a perfect sequence match between the read and the allele). Finally, the reads are assigned to the allele with which they have the highest similarity score.

LRcaller. LRcaller considers each SV breakpoint separately. It extracts the sequence contained in a 1kb window centered around a given breakpoint for both the reference and alternative alleles. The selected reads are cropped based on their alignment to the reference genome, so that the obtained sequence is contained in the breakpoint window. The cropped reads are then aligned to both allele breakpoint

sequences using the global alignment function of SeqAn (Döring et al., 2008), and assigned to the allele with which they have the highest similarity score. LRcaller also proposes three other genotyping methods that assign reads to alleles based on an alignment distance against the reference sequence of the SV breakpoints, computed from the input alignment to the reference. A final genotyping method combines both approaches of re-alignment to the allele sequences and distance to the reference.

SVJedi. On the other hand, the reads selection of SVJedi does not rely on a preliminary step of read mapping to the reference genome. It takes as input the original sample of long reads, along with a VCF describing the SVs and the sequence of the reference genome. SVJedi first represents the sequence of the reference and alternative alleles for each SV, using a window extending 5 kb over the breakpoint positions. It then maps the long reads to the collection of allele sequences for the whole SV set using minimap2 (Li, 2018). This mapping approach acts as a preliminary selection and partitioning of the reads. The reads are further selected to ensure that they provide confident information on the genotype, by requiring their alignment to cover a 200 bp window centered around the breakpoint position and to have a minimum mapping score (MAPQ).

2.3 Current limits of structural variant genotyping

Reference bias. A predominant limit of most SV genotyping methods is the reference-bias of the genotype inference introduced by the preliminary read-mapping to the reference genome. It is particularly the case for the force-calling approach, where alternative allele-supporting reads are inferred purely based on their alignment to the reference genome. While VaPoR and LRcaller methods limit this bias by realigning the reads to both reference and alternative alleles sequences for the read partition, the initial information of the mapping location of reads is still given by the read mapping to the reference only. Meaning that there may be reads that inherently support the alternative alleles of SV in the sequenced sample ending up unused for the genotype inference because they are unmapped or inaccurately mapped on the reference genome during the first step. This issue is fully tackled by SVJedi by mapping reads directly to both alleles, ensuring an equal weight of both alleles in the read selection and count process.

Close and overlapping SVs. The linear representation of SV alleles performed by SVJedi, as well as VaPoR and LRcaller, presents its own limit. In the case of genomic regions harbouring close or overlapping SVs, the accurate representation of each individual SV alleles is not trivial, as it ideally should take into account all

possible combinations of neighbouring SV alleles. Additionally, the length of the window's overrun at each side of the SV's breakpoints in the allele representation can greatly impact the mappability of informative reads, but an optimal value for this parameter is difficult to determine. An intuitive solution to this issue is to move from a linear to a graph-based representation, such as the variation graph data structure introduced by Paten et al. (2017) and Garrison et al. (2018). Currently, the use of a variation graph to represent SV alleles has been developed for variant genotyping in Paragraph (Chen et al., 2019), graphTyper2 (Eggertsson et al., 2019), the latest Giraffe (Sirén et al., 2021), and PanGenie (Ebler et al., 2022). These four tools all use short reads to infer genotypes, and there is no variation graph-based genotyping tool using long reads.

3 Pangenome graphs: a new era for the characterization of SVs

It is now accepted by the community that the traditional linear representation of reference genomes induces an issue of reference-bias for SV discovery and genotyping. Whether they represent one individual, or even one mosaïc haplotype, they lack in representing the genome variability of species needed to obtain an optimal mapping of re-sequencing data.

A solution to overcome this limitation was proposed in the form of variation graphs. Although the idea of comparing genomes through graphs was already in place with Cactus (Paten et al., 2011), the task of multiple genome alignment over a large collection of genomes is computationally intensive. This computational cost was reduced by the use of trees to guide the genome alignments in Progressive Cactus (Armstrong et al., 2020), but this made it unsuitable to use as a way to represent intra-specific diversity, as there is not a single tree that can reflect the evolution history of the whole genome. Variation graphs, first introduced by Garrison et al. (2018) with their VG toolkit, are sequence graphs aiming to represent the full sequence variability in the genomes of a species. VG achieved a fast graph construction by building its graph from one input reference genome and a given VCF set of known variants. This idea was quickly followed by the introduction of pangenome graphs (Li et al., 2020; Garrison et al., 2023; Hickey et al., 2024), which are a type of variation graph built directly from a collection of genomes through whole genome alignment. The current pangenome graph tools employ different strategies to construct a graph while avoiding the cost of multiple genome alignment. They are described in Section 3.2.

Variation graphs and pangenome graphs were shown to improve mapping and SNP calling with short reads (Garrison et al., 2018; Hickey et al., 2024), allowing

the discovery of new genomic variants in populations and a higher quality of SV genotyping.

3.1 Advantages of pangenome graphs over variation graphs

The key difference between pangenome graphs and variation graphs (*i.e.* graphs built with VG toolkit) is their input requirement. While variation graphs are faster to construct because they do not perform whole genome alignment, the accuracy and completeness of the genomic variation they include depend directly on the quality of the provided VCF. Additionally, variation graphs are still subject to the issue of reference bias, as they can not represent a variant in a region absent from the reference genome (*e.g.* in an insertion). Pangenome graphs on the other hand, by aligning multiple genomes rather than relying on an input VCF, are not troubled with the issue of SV caller-induced false positives or false negatives, nor with the problem of multi-sample VCF merging.

The past few years have seen an increasing use of pangenome graphs to study SVs in various species such as wheat, human, chicken, cow, grape or pig (Bayer et al., 2022; Liao et al., 2023; Rice et al., 2023; Jang et al., 2023; Cochetel et al., 2023; Miao et al., 2024).

3.2 Pangenome graph construction

The different pangenome graph constructing tools, namely Minigraph (Li et al., 2020), Minigraph-Cactus (Hickey et al., 2024), and PGGB (Garrison et al., 2023), all rely on whole genome alignment. However, they have significant differences in their approach to represent the variants and to construct the graphs. For instance, Minigraph and Minigraph-Cactus rely on a user-designated reference-genome to guide the structure of the graph, while PGGB aims to construct a reference-free graph. However, they all take a collection of genomes as input and output a graph in the GFA format.

Minigraph. Minigraph borrowed the minimizer mapping approach from minimap2 (Li, 2018) and enhanced it to map sequence on a graph. From the alignments, Minigraph is able to produce a new graph including new variants. In order to construct a pangenome graph from scratch with a collection of genomes, it first initializes the graph from the alignment of a first sample genome to the designated reference genome. It then updates the graph iteratively with each remaining sample genome. For the purpose of efficient read mapping, Minigraph only considers SVs from 100 bp to 100 kbp and tries to keep the final graph as linearized as possible. Moreover, it is suited for intra-specific or closely related genomes comparison.

An interesting feature of Minigraph is its usability for long read mapping to a pangenome graph.

Minigraph-Cactus. Minigraph-Cactus is based on the combination of Minigraph and the multiple genome comparative tool Progressive Cactus (Armstrong et al., 2020). Its pipeline can be summarized in three main steps. Firstly, the backbone of the graph is constructed with Minigraph. Secondly, the genomes are realigned to the graph using the Minigraph mapper and Progressive Cactus, which allows to locally identify and add smaller variants to the graph. Finally, the graph is polished and unmapped segments are removed. In addition to the GFA file containing the graph, Minigraph-Cactus also outputs a VCF containing the variants represented in the graph using vg deconstruct (Liao et al., 2023). As Minigraph, Minigraph-Cactus is mainly intended to construct intra-species pangenome graphs. The release of Minigraph-Cactus also came with a module to convert the Progressive Cactus output file into a pangenome-like graph in GFA format. This allows for the construction of inter-species pangenome graphs, as Progressive Cactus is mainly intended for the comparison of divergent species, using a species tree to guide the multiple genome alignments.

Pangenome Graph Builder (PGGB). In order to produce an 'unbiased' pangenome graph, the PGGB pipeline performs $n * n - 1$ pairwise genome alignments with the n input genomes using the wfmash aligner[3]. It then uses seqwish (Garrison and Guarracino, 2023) to transform the collection of pairwise alignments into a multiple alignment, which it then translates into a variation graph. This graph undergoes two polishing steps, in order to smooth the graph and remove the sequence redundancy. Like Minigraph-Cactus, PGGB outputs a VCF reporting the variants in the graph using vg deconstruct (Liao et al., 2023).

Both Minigraph-Cactus and PGGB represent all variant sizes (SNPs, indels, SVs) in their graph. They are heavy pipelines comprising many steps, some of which are very little described (*e.g.* the smoothxg tool[4] used in one of the graph polishing step of PGGB). Furthermore, they are quickly evolving, so much as so the version of their pipelines described in their respective papers is already outdated.

3.3 Structural variants in pangenome graphs

Detecting the variants represented in pangenome graphs comes down to finding 'bubbles' in the graphs. Detecting bubbles in graphs has been a question since

[3] https://github.com/waveygang/wfmash
[4] https://github.com/pangenome/smoothxg

before the advent of pangenome graphs (Dabbaghie et al., 2022). However, as pangenome graphs are novel types of graphs with a notably high number of bubbles, and as variant bubble detection calls for specific information about the bubbles (*e.g.* base position of the bubbles in the genomes), dedicated tools for bubble detection in pangenome graphs were developed.

VG deconstruct. VG deconstruct (Liao et al., 2023) reports variant bubbles in a VCF format, described with the standard variant information found in VCFs (*e.g.* base position relative to a chosen reference, sequence of alleles) along with graph-specific information (*e.g.* identifiers of the nodes contained in the bubbles, path of the sample genomes through the bubbles). As vg deconstruct reports all variant bubbles, its output VCF files contain small variants along with the SVs if represented in the input pangenome graph.

Gfatools. Gfatools (Li, 2019) reports variant bubbles in a BED format, described by their base position relative to a chosen reference, their base and node lengths. Gfatools only accepts a specific version of the GFA format (rGFA), which is only produced by minigraph.

VG deconstruct and Gfatools have indeed been applied to characterize variants in bovine (Leonard et al., 2023), wild grape (Cochetel et al., 2023) and human (Liao et al., 2023) pangenome graphs studies. Unfortunately, the methods of the tools can not be compared, as neither tool provides a description of their bubble detection method.

3.4 Current limits of SV characterization in pangenome graphs

Annotation of SV type. Although vg deconstruct and gfatools detect the SVs bubbles in pangenome graphs, they do not annotate the variant type of the bubbles. While recognizing SNP bubbles can be pretty straightforward, as they are expected to have exclusively 1 bp path lengths, discriminating between different types of SVs and non-SV bubbles (*i.e.* bubbles resulting from highly divergent regions difficult to align) is not immediate. Two tools, vcfbub and vcfwave, from the vcflib suite (Garrison et al., 2022), perform variant annotation on graph-produced VCFs. Vcfbub unravels the nested bubbles from an input VCF to only keep the highest level variants, and vcfwave realigns the variant alternative alleles to the reference allele to annotate the variants - with their type - contained in the higher level bubbles.

SV genotyping from re-sequencing samples. The pipelines of Minigraph-Cactus and PGGB can output the genotypes of the detected SVs for the genomes contained in the graph. However, genotyping the SVs in populations sequenced with short and long reads and directly mapped on a pangenome graph is also required. Pangenie (Ebler et al., 2022) is a mapping-free variant genotyper with short-reads, that is able to use a VCF obtained from *vg deconstruct*. Giraffe (Sirén et al., 2021) is a read mapper on variation graph, and was also reported to perform SV genotyping in Ebler et al. (2022) and Liao et al. (2023) papers, but the corresponding method has not yet been described or published.

III Long-Read SV Genotyping On A Variation Graph

In this chapter

1 Introduction 35
Paper: SVJedi-graph: improving the genotyping of close and overlapping structural variants with long reads using a variation graph 36

1 Introduction

During the beginning of my thesis, I worked on the question of long-read SV genotyping using a type of sequence graph, the variation graph, to represent the genome along with the alternative alleles of the SVs. The long read sequencing data of the *Coenonympha* butterflies were not yet available at this time, but graphs seemed like an appealing approach to compare the *Coenonympha* genomes, as it did not require to choose a reference genome between the two parental species. This work thus allowed me to explore how SVs could be represented in such graphs, in addition to resolving the limit of close SV genotyping identified in the state of the art and in particular in SVJedi, a tool developed by my team.

As mentioned in Chapter II (Section 1.1), the advent of long read sequencing technologies revolutionized the detection of SVs in genomes, improving both its recall and its accuracy. This brought back to light their contribution to genetic diversity, and their association to phenotypic variability, species adaptation and evolution, and clinical disorders. When analyzing and comparing SVs between individuals, a crucial problem is their accurate genotyping. All state of the art methods dedicated to SV genotyping with long-read data are limited either by their unequal representation of the different SV alleles leading to a reference-bias, or by their linear representation of the alleles hindering the genotyping of close or overlapping SVs. In response to this, I developed a new method that relies on a variation graph to represent all alleles from a set of SVs into a single data structure. It uses a state of the art sequence-to-graph mapper to map the long reads on

the variation graph, and estimates the most likely genotype for each SV from the calculated mapping depth on the allele-specific edges in the graph. By applying SVJedi-graph on simulated sets of close and overlapping deletions, I demonstrated that this graph model overcomes the reference-bias issue while maintaining a high genotyping accuracy whatever the SV proximity. On the human gold standard HG002 dataset from Genome in a Bottle (Zook et al., 2020), SVJedi-graph obtained the best performances among long-read SV genotypers, genotyping 99.5% of the high confidence SV callset with an accuracy of 95% in less than 30 minutes of time.

This chapter is presented in the form of a paper, which was published in the journal *Bioinformatics* (Romain and Lemaitre, 2023). I presented an early version of this work at the *Data Structures in Bioinformatics* (DSB) workshop event of 2022 in Düsseldorf (Germany), and its paper version at the *Intelligent Systems For Molecular Biology* and *European Conference on Computational Biology* (ISMB/ECCB) joint conference event of 2023 in Lyon (France). I implemented SVJedi-graph in Python, and still maintain it on GitHub[1] and as a BioConda package[2].

[1] https://github.com/SandraLouise/SVJedi-graph
[2] https://anaconda.org/bioconda/svjedi-graph

Bioinformatics, 2023, **39**, i270–i278
https://doi.org/10.1093/bioinformatics/btad237
ISMB/ECCB 2023

SVJedi-graph: improving the genotyping of close and overlapping structural variants with long reads using a variation graph

Sandra Romain[1] **and Claire Lemaitre** [1,*]

[1]Univ Rennes, Inria, CNRS, IRISA, Rennes F-35000, France

*Corresponding author. University of Rennes, Inria, IRISA, Rennes F-35000, France. E-mail: claire.lemaitre@inria.fr

Abstract

Motivation: Structural variation (SV) is a class of genetic diversity whose importance is increasingly revealed by genome resequencing, especially with long-read technologies. One crucial problem when analyzing and comparing SVs in several individuals is their accurate genotyping, that is determining whether a described SV is present or absent in one sequenced individual, and if present, in how many copies. There are only a few methods dedicated to SV genotyping with long-read data, and all either suffer of a bias toward the reference allele by not representing equally all alleles, or have difficulties genotyping close or overlapping SVs due to a linear representation of the alleles.

Results: We present SVJedi-graph, a novel method for SV genotyping that relies on a variation graph to represent in a single data structure all alleles of a set of SVs. The long reads are mapped on the variation graph and the resulting alignments that cover allele-specific edges in the graph are used to estimate the most likely genotype for each SV. Running SVJedi-graph on simulated sets of close and overlapping deletions showed that this graph model prevents the bias toward the reference alleles and allows maintaining high genotyping accuracy whatever the SV proximity, contrary to other state of the art genotypers. On the human gold standard HG002 dataset, SVJedi-graph obtained the best performances, genotyping 99.5% of the high confidence SV callset with an accuracy of 95% in less than 30 min.

Availability and implementation: SVJedi-graph is distributed under an AGPL license and available on GitHub at https://github.com/SandraLouise/SVJedi-graph and as a BioConda package.

1 Introduction

Structural variants (SVs) are genomic rearrangements of at least 50 bp that differ between the genomes of individuals belonging to the same species. This definition encompasses a wide range of variations in terms of size and type. The most frequent types are deletions and insertions, but there are also balanced SVs such as inversions and translocations. Although SVs are less frequent in numbers than punctual variations, they often involve more base pairs in the genomes and have long been shown to be involved in phenotypic variability, species adaptation and evolution, and in many diseases and disorders (Weischenfeldt et al. 2013; Mahmoud et al. 2019).

With the democratization of long-read sequencing technologies, there has been an increasing number of studies focusing on the characterization and analysis of this type of genetic variation on a genome-wide scale in various organisms. Indeed, because of their large size and frequent localization in repeated regions, SVs were very challenging variants to identify with short reads (Mahmoud et al. 2019; Delage et al. 2020). Long reads have really changed the game in this field, allowing their reliable and accurate detection in resequencing genome data (Chaisson et al. 2019; Zook et al. 2020). In particular, in recent years, numerous studies have been conducted at the population level with large sample sizes, revealing associations between SVs and phenotypes of interest or their involvement in changes in gene expression in various organisms, such as, for example, in plants (Alonge et al. 2020), in yeasts (O'Donnell et al. 2022) and of course in human populations (Beyter et al. 2021; Porubsky et al. 2022), to cite only a few of them.

In most of these studies, the input of the analyses is typically a matrix with variants in lines and samples or individuals in columns (or *vice versa*) containing in each cell the genotype or number of each allele of the given variant in the given individual. To obtain such a matrix, the commonly accepted approach is composed of two steps: the first one consists in obtaining a most comprehensive and nonredundant set of SVs. This is achieved by using SV discovery tools on all or a subset of the samples to identify all structural variants in samples compared to a reference genome. The obtained call sets are then merged to obtain a nonredundant set of SVs which defines the lines of the matrix. Then, the second step is the genotyping and aims at filling the matrix with genotypes. Genotyping one variant in an individual consists in counting how many reads from this individual support each described allele of the given variant. Based on these read counts, a genotype is derived, typically homozygous for the reference or alternative allele or heterozygous for bi-allelic variants in a diploid individual. In such a genotyping step, all samples are thus evaluated through the same SV call set to obtain comparable values.

Genotyping and discovery are therefore two distinct tasks that necessitate different methods and we have witnessed a strong increase in the number of tools developed in recent years dedicated purely to the genotyping problem. While genotyping appears as a simpler problem than discovery, since variants are already known and the whole reference genome

does not need to be blindly investigated, there are some issues that deserve special attention. The first issue is named the reference bias. It is well known that the more similar two sequences are, the easier it is to align them and this is emphasized in the context of structural dissimilarity and with reads containing many sequencing errors. Therefore, when mapping reads only to the reference genome, one may favor the reference allele. As the different alleles are well defined in the genotyping problem, the reference bias should be avoidable, by mapping the reads on both reference and alternative alleles in an equal manner. This can be achieved by generating allele-specific subsequences and mapping the reads only on these sequences. Typically, these sequences are centered on the SV breakpoints and include neighboring sequences of size depending on the average read size.

The second issue concerns closely located or overlapping SVs for which representing the sequences of the different alleles is not so trivial. For a given SV, the neighboring sequences are not uniquely defined as they depend on the allele states of neighboring SVs. Intuitively, moving from a linear to a graph-based representation solves this issue. In a variation graph, each node is a sequence, edges represent adjacencies between sequences observed in an allele and each combination of alleles is represented by a path in this graph. As a matter of fact, numerous implementations for building variation graphs or also named pangenome graphs, analyzing such graphs or mapping reads on them are now available and commonly used (see Paten et al. 2017; Garrison et al. 2018; Li et al. 2020, Rautiainen and Marschall 2020; Guarracino et al. 2022 to cite only a few). Importantly, they have been shown to improve read mapping and small variant genotyping (Eggertsson et al. 2017; Garrison et al. 2018). As concerns SVs, the last generation of SV genotypers for short Illumina reads, Paragraph (Chen et al. 2019), graphTyper2 (Eggertsson et al. 2019), the latest Giraffe (Sirén et al. 2021), and PanGenie (Ebler et al. 2022), are all based on such sequence graphs.

As genome sequencing is more and more achieved with long-read technologies, even for population scale studies (Coster et al. 2021), and because the large size of the reads has an undeniable benefit on the quality of SV analyses (for genotyping as well as for discovery) (Mahmoud et al. 2019, Duan et al. 2022), a few tools dedicated to SV genotyping with long-read data have been proposed these last years, namely VaPoR (Zhao et al. 2017), SVJedi (Lecompte et al. 2020), and LRcaller (Beyter et al. 2021). However, none of them uses a graph representation of the variants. All three tools explicitly represent the allelic sequences, but as linear sequences, and map the reads on both reference and alternative allele sequences. However, only SVJedi strictly avoids the reference bias by mapping all the reads on all allelic sequences, whereas VaPoR and LRcaller perform first a selection of the reads to be mapped on alleles based on a whole reference genome alignment given as input. Additionally, Sniffles (Sedlazeck et al. 2018) and CuteSV (Jiang et al. 2020), which are discovery tools, also provide a genotyping mode as an option, but the methods implemented for these optional modes have not been described in any publication. As these tools require a mapping on the reference genome as input, we can hypothesize that they mainly rely on the split-read signatures used also in their discovery mode and may thus be subject to the reference bias.

We present here the first SV genotyper for long-read data that is based on a variation graph. By avoiding the mapping on the reference genome only and using a variation graph representing the whole reference genome complemented by all described alternative alleles given in the input SV call set, our method is not reference-biased and improves the genotyping of distant, as well as closely located and overlapping SVs.

2 Materials and methods

Our method relies on the representation of structural variants with a variation graph, which is then used as "reference" to map the long reads on.

It takes as input the set of SVs to genotype in VCF format, the sequence of the reference genome in FASTA format, and the long reads from which the SV genotypes will be estimated in FASTQ or FASTA format (compressed or not). The main output is a VCF file, corresponding to the input VCF file with an additional column containing the predicted genotypes of the SVs. It also outputs the variation graph representing the whole genome and alternative alleles of the input SVs in GFA format.

Our method is composed of four steps, illustrated in Fig. 1. First, we build the variation graph from the reference genome and the SV set. We then use an external tool, minigraph (Li et al. 2020), to map the long reads on the graph we produced. The alignment results are filtered to identify genotype-informative reads, which are stored by covered SV and supported allele. Finally, the read counts are normalized and we attribute the genotype with the maximum likelihood to each SV of the input set.

2.1 Constructing the variation graph

A variation graph is a directed graph whose nodes are labeled with nonoverlapping genomic sequences. Edges represent sequence adjacencies observed in a genome or allelic sequence. A path in the graph represents a possible haplotype in a genome.

In our method, we construct a variation graph from the sequence of the reference genome and a set of SVs characterized by their type, their breakpoint positions on the reference genome, and their sequence in the case of insertions. The first step is to list and sort all breakpoint positions for each chromosome of the reference genome, then use them in the second step to fragment the reference sequence of the chromosome into reference nodes. Reference edges are added between each pair of successive reference nodes, forming the path of the reference genome in the graph.

The third step is to add additional edges for each SV described in the input VCF according to its type to form the path of the alternative allele. We call such edges alternative edges. In the case of insertions, an alternative node is also added, labeled with the sequence of the insertion. Thus, in our variation graph, all edges represent breakpoints of the input SVs, that is sequence adjacencies that are specific to one of the alleles.

The resulting variation graph is output in the GFA format.

In our graph, we currently can represent deletions, insertions, inversions, and intra-chromosomal translocations. The first step of the construction allows for the representation of overlapping SVs.

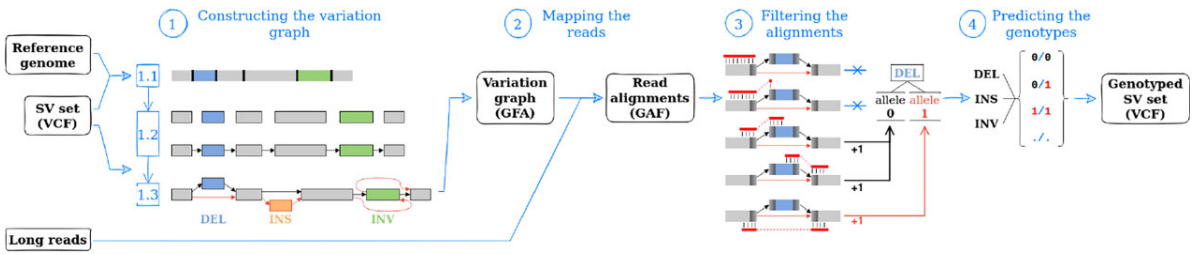

Figure 1. Illustration of the four steps of SVJedi-graph. The method takes three files as input: the sequence of the reference genome, the VCF describing the SVs to genotype, and the long reads to genotype the SVs from. The first step is the construction of the variation graph, the second step is the mapping of the long reads on the variation graph with minigraph (producing the GAF alignment file), the third step is the filtering of the reads, and the final fourth step is the genotype prediction. Two files are output, with the main one being the genotyped version of the input VCF, and the other one being the GFA containing the variation graph.

2.2 Mapping the reads on the graph

The long reads are mapped on the constructed variation graph with minigraph (v0.19) (Li et al. 2020), with the "-x lr" argument for aligning long reads and without base-level alignment to increase speed since only the read position on the graph is needed in our method to predict the SV genotype. Minigraph outputs the alignments' results in the GAF format, which is a variation of the PAF format adapted to sequence graphs.

We have also tested another mapper, GraphAligner (Rautiainen and Marschall 2020) and the base-level alignment mode of minigraph. We chose minigraph without base-level alignment which gave the best results.

2.3 Selecting the informative reads

In our method, we consider that a read aligning on at least one breakpoint sequence of an SV gives information on that SV's genotype, since breakpoints are sequence adjacencies specific to one or the other SV allele. Each SV has one or two breakpoints for each of its alleles depending on its type. For example, deletions have two breakpoints for their reference allele, that we will call reference breakpoints, and one breakpoint for their alternative allele, that we will call alternative breakpoint. Inversions have two reference breakpoints and two alternative breakpoints. Each breakpoint is represented by a distinct edge in the variation graph. For each alignment output by minigraph, we first verify that the read aligns on at least two nodes, meaning that it overlaps at least one of the breakpoints in the graph. Then, we list all SVs that have at least one breakpoint included in the span of the read alignment on the graph. For each of the listed SVs, we determine which allele is covered by the alignment and increment by one the support value for this SV's allele for each allele breakpoint that the read alignment covers. A read is considered as covering a breakpoint if the alignment overlaps at least d_{over} base pair from each side of the breakpoint. We fixed the default value of d_{over} at 100 bp.

Each time a read is counted as allele support for an SV, the support count for this SV's allele is incremented by one in a dictionary containing the SVs as keys. If a read covers both breakpoints of an SV allele, it counts as two read supports.

2.4 Predicting the SV genotypes

Once all the alignments produced by minigraph have been processed, the genotype of each SV is estimated using the read support counts for its alleles. First, for SV types with an unbalanced number of breakpoints between alleles (deletions and insertions), the support count is normalized for each allele by the allele's breakpoint number (e.g. for a deletion, the reference allele count is divided by two). Then, the normalized allele support counts are used to compute the likelihood for each possible genotype in a diploid individual (homozygous for reference 0/0, heterozygous 0/1, or homozygous for alternative 1/1). We use the same likelihood formula as in SVJedi and CuteSV (Lecompte et al. 2020; Jiang et al. 2020), which is described in Nielsen et al. (2011). Basically, the likelihoods of the three possible genotypes given the observed normalized read counts (c_0 and c_1 for reference and alternative alleles, respectively) are computed based on a simple binomial model:

$$\mathcal{L}(0/0) = (1-p_e)^{c_0} \times p_e^{c_1} \times C^{c_0}_{c_0+c_1} \quad (1)$$

$$\mathcal{L}(1/1) = p_e^{c_0} \times (1-p_e)^{c_1} \times C^{c_0}_{c_0+c_1} \quad (2)$$

$$\mathcal{L}(0/1) = \left(\frac{1}{2}\right)^{c_0+c_1} \times C^{c_0}_{c_0+c_1} \quad (3)$$

where p_e is the probability that a read maps to a given allele erroneously, assuming it is constant, and independent between all observations. p_e was fixed to 5×10^{-5}, after empirical experiments. The genotype with the largest likelihood is assigned and all three likelihoods are also output (−log10 transformed) as additional information in the VCF file.

Finally, we report the genotype of an SV only if it is supported by a minimal amount of supporting reads (sum of allele counts after normalization), otherwise a missing genotype ("./.") is reported. This is governed by a user-defined parameter, whose default value is set to 3.

2.5 Implementation

The presented method is implemented in Python under the name SVJedi-graph (v1.1.1) and is available on github (https://github.com/SandraLouise/SVJedi-graph) and as a conda package (https://anaconda.org/bioconda/svjedi-graph). Currently, SVJedi-graph can genotype five types of SVs: deletions, insertions, duplications, inversions, and intra-chromosomal translocations. Insertions need to be sequence-resolved with the full inserted sequence characterized and reported in the ALT field of the VCF file. As duplications are a special case of insertions, SVJedi-graph supports also duplications, as long as their duplicated sequence is characterized and reported similarly to insertions.

2.6 Simulating close and overlapping SV datasets

In order to evaluate our method's genotyping performances on closely located or overlapping SVs, we simulated twelve deletion datasets with varying distance ranges between

deletions on the human chromosome 1 (assembly GRCh37.p13). All those datasets shared the same 995 deletions selected from the dbVar database (Phan et al. 2017), ranging from 50 bp to 10 kb in size and distant of at least 10 kb from each other. These deletions were equally distributed over the three possible genotypes (0/0, 0/1, 1/1), resulting in two synthetic haplotype sequences of chromosome 1. These haplotype sequences were used to simulate a single long-read sequencing dataset using simLoRD (Stöcker et al. 2016), with a PacBio error profile and error rate of 16% and at a sequencing depth of 30×.

Then, we generated 12 different variant sets (VCF files) by adding to each of those 995 "initial" deletions one simulated deletion at different distance or overlapping ranges from its companion deletion. The size of these additional deletions ranged from 50 bp to 2 kb, and they were all recorded as homozygous reference genotype (0/0) in the variant file, meaning that these additional deletions are not present in the simulated sequenced individual. Therefore, the same set of simulated reads can be used to genotype the different deletion sets. Six of the deletion sets correspond to nonoverlapping deletions, with random distance of: (i) 5–10 kb, (ii) 1–5 kb, (iii) 500–1 kb, (iv) 100–500 bp, (v) 50–100 bp, and (vi) 0–50 bp. They contain each 1990 deletions with a median size around 1 kb. The other six sets simulated overlapping deletions, with random overlapping of: (i) 0–50 bp, (ii) 50–100 bp, (iii) 100–200 bp, (iv) 200–300 bp, (v) 300–400 bp, and (vi) 400–500 bp. The size of the sets varies depending on the overlap size range, since only the deletions larger than the minimal overlap bound were kept. These overlapping deletion sets contain between 1382 (400–500 bp overlaps) and 1990 deletions (0–50 bp overlaps). Accordingly, deletions are larger in the sets with the largest overlap sizes (the median deletion size ranges from 1 to 1.3 kb). All simulated datasets are available for download (see Supplementary Material).

2.7 Evaluation and comparison to state of the art long-read genotypers

We evaluated and compared our method to other genotypers on its genotyping quality and computing performances. To evaluate the genotyping quality, we used two metrics: the genotyping accuracy and the genotyping rate.

We define the genotyping rate as the percentage of input SVs for which the tool was able to attribute a genotype. It was calculated using Equation (4), where TP is the number of SVs for which the predicted genotype corresponds to the true genotype, FP is the number of SVs for which the predicted genotype differs from the true genotype, and FN is the number of SVs that could not be attributed a genotype. We define the genotyping accuracy as the percentage of genotyped SVs that were attributed their true genotype, calculated with Equation (5).

$$\text{Genotyping rate} = \frac{TP + FP}{TP + FP + FN} \times 100 \quad (4)$$

$$\text{Genotyping accuracy} = \frac{TP}{TP + FP} \times 100. \quad (5)$$

We compared our method to four state of the art long-read SV genotypers, namely SVJedi (Lecompte et al. 2020) (v1.1.6), cuteSV (Jiang et al. 2020) (v1.0.13), Sniffles2 (Sedlazeck et al. 2018) (v2.0.6), and LRcaller (Beyter et al. 2021) (v1.0). SVJedi and LRcaller are tools dedicated to SV genotyping, while cuteSV and Sniffles2 are primarily SV callers. CuteSV and Sniffles2 were run with their "force call" option to genotype a given set of SVs, bypassing the SV discovery steps. For each comparison, all tools were run with the same variant file as input. SVJedi performs the read mapping internally using minimap2, while cuteSV, Sniffles2, and LRcaller take as input the results of the read mapping done externally. SVJedi was run on the ONT reads with the parameter "-d ont," and on both PacBio datasets with the default parameter "-d pb." As SVJedi uses minimap2 (v2.17), we also used minimap2 (Li 2018) (v2.17) to map the reads on the reference genome, as input to cuteSV, Sniffles2, and LRcaller. Minimap2 was run on the PacBio CLR reads, PacBio HiFi reads, and ONT reads with the parameters presets "map-pb," "asm20," and "map-ont," respectively. LRcaller has five methods to genotype SVs, we used the joint method as done in the benchmark paper for SV genotyping methods (Duan et al. 2022) with the argument "–gtm joint." All tools but Sniffles2 were run using 20 CPU threads. Command lines used to run the tools are given in Supplementary Material.

3 Results

3.1 Impact of SV proximity in simulated datasets

In order to evaluate the benefits of using a variation graph to genotype SVs, and in particular close and overlapping SVs, we first applied our method to several simulated datasets of deletions in the human chromosome 1, in which we controlled the distance or overlap size of consecutive pairs of deletions (see Section 2).

Figure 2 shows the performance metrics of SVJedi-graph and the other compared genotypers as a function of the distance between pairs of consecutive deletion segments. We

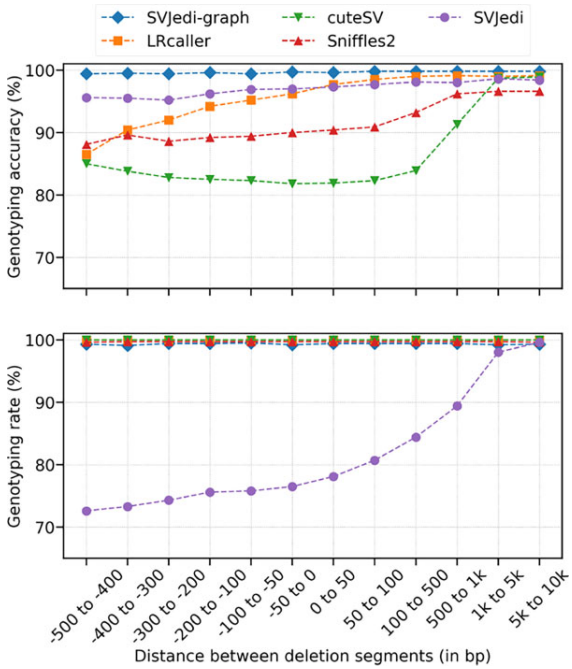

Figure 2. Genotyping performances of long-read SV genotypers on the 12 simulated deletion datasets on human chromosome 1, with varying distances between pairs of consecutive deletions. The *X* axis represents the different simulated datasets ordered by increasing distance between pairs of consecutive deletion segments. Negative *X* values correspond to datasets with overlapping deletions.

observe that the distance between deletions and the fact that some deletions overlap each other does not impact SVJedi-graph performances. It maintains very high accuracy and rate (above 99%) whatever the distance between the simulated deletions and even for overlapping deletions.

On the contrary, all the other tested genotypers show decreasing genotyping qualities when the deletions are closer to one another. CuteSV and Sniffles genotyping accuracy starts decreasing as soon as deletions are <1000 bp apart, falling ~80% and 90%, respectively, for overlapping or adjacent deletions. The drop in accuracy is smaller for LRcaller, which maintains its accuracy above 97% for even very close deletions, but falls below 90% for overlapping deletions with the largest overlaps. On the other hand, SVJedi maintains its high accuracy but its genotyping rate decreases regularly with the deletion proximity. It is not able to assign a genotype to more than 20% of the deletions that are <50 bp apart.

3.2 Impact of breakpoint position precision in simulated datasets

In practice, real SV call sets may not be defined at the base pair resolution and can contain breakpoint positions shifted from the real positions. In order to evaluate to what extend this imprecision in breakpoint definition may impact the genotyping performances, we applied the genotypers on the previous simulated long-read dataset but with imprecise input VCF files, where the positions of the 995 deletions used to simulate the reads were shifted. Both breakpoints of the deletions were shifted by a fixed distance in the same direction to preserve the deletion size and we retained only those deletions where the deleted segment overlapped by at least 50% with the deletion from which it was moved.

Figure 3 shows the performance metrics of SVJedi-graph and the other compared genotypers when increasing the breakpoint shift from 10 to 1000 bp. Except for cuteSV which remarkably seems not to be impacted by breakpoints shifts, all other genotypers show decreasing genotyping accuracies when the imprecision increases. SVJedi-graph still maintains a high accuracy as long as the imprecision is smaller than 200 bp (98.8% for 200 bp breakpoint shifts).

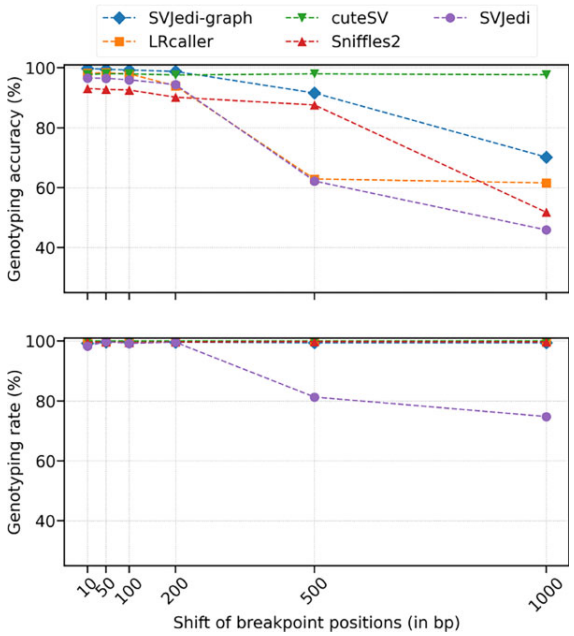

Figure 3. Genotyping performances of long-read SV genotypers on simulated deletion datasets on human chromosome 1 with varying levels of imprecision in the breakpoint definitions. The *X* axis represents the distance in base pairs between the deletions breakpoints as simulated in the input read dataset and their corresponding shifted breakpoints given in the different input VCF files.

3.3 Results on real human benchmark datasets

To assess genotyping accuracy on real data, one needs a comprehensive set of well characterized SVs with their genotype well ascertained in at least one individual. The consortium Genome in a Bottle (GIAB), thanks to massive data production and manual efforts, produced such a dataset dedicated to SV tools benchmarking on the human individual HG002, son of the so-called *Ashkenazi trio* (Zook et al. 2020). This highly curated set, referred as High confidence, contains 5464 deletions and 7281 insertions of at least 50 bp which are distant from one another from at least 1 kb, whose genotypes are heterozygous or homozygous for the alternative allele in HG002 (Tier 1 set v.0.6 with the tag "PASS" in the VCF FILTER field) We applied SVJedi-graph on this set with three long-read datasets from the HG002 individual obtained with different sequencing technologies, namely PacBio CLR, PacBio HiFi and Nanopore (ONT), and provided by GIAB (see Supplementary Material for download links).

With the 30× CLR PacBio reads, SVJedi-graph was able to genotype 99% of the SVs with a genotyping accuracy of 94.6% (Tables 1 and 2). This accuracy is slightly better for deletions than for insertions for this set (1.3 point %), as well as the genotyping rate (1.6 point %). Table 1 shows a contingency table of the obtained genotypes compared with expected ones for deletions and insertions. Most genotyping errors happen on alternative homozygous (1/1) deletions or insertions that end up predicted as heterogygous (0/1). Out of the errors made on alternative homozygous variants, 99% and 98% were wrongly predicted heterozygous deletions and insertions, accounting for 56% and 77% of all genotyping errors for each of these SV types, respectively. For heterozygous deletions, the errors are well balanced between wrongly genotyped reference homozygous and alternative homozygous, especially for insertions (50% for each of the two

Table 1. Contingency tables of SVJedi-graph genotyping results on the real 30× PacBio dataset of human individual HG002 with respect to the high confidence GIAB call set.[a]

Deletions

		SVJedi-graph predictions			
		0/0	0/1	1/1	./.
GIAB	0/1	34	3322	75	2
	1/1	2	142	1884	3

Insertions

		SVJedi-graph predictions			
		0/0	0/1	1/1	./.
GIAB	0/1	47	3397	46	15
	1/1	6	324	3339	107

[a] Results for the 5464 deletions (top) and 7281 insertions (bottom) are indicated in two separated tables, where columns indicate SVJedi-graph genotypes and rows GIAB ones. Gray labeled boxes, in the diagonal, give the amount of variants correctly genotyped by SVJedi-graph. The number of genotypes that SVJedi-graph fails to assess is indicated by the "./." column.

Table 2. Genotyping accuracy and rate of SVJedi-graph and state of the art genotyping tools on the deletions and insertions of the *High confidence* HG002 SV set with the 30× CLR PacBio read dataset.

	Global		Deletions		Insertions	
Tool	Accuracy (%)	Rate (%)	Accuracy (%)	Rate (%)	Accuracy (%)	Rate (%)
SVJedi-graph	94.6	99.0	95.4	99.9	94.1	98.3
cuteSV	88.7	100	90.8	100	87.1	100
LRcaller	83.6	100	89.3	100	79.3	100
Sniffles2	85.4	99.5	87.9	99.9	83.6	99.1
SVJedi	92.2	90.2	91.7	85.8	92.5	93.6

possible wrong genotypes). For heterozygous deletions, 69% of the errors were alternative homozygous genotype predictions.

The four other tested genotyping tools present a lower accuracy than SVJedi-graph on the global SV set (92.2%, 88.7%, 85.4%, and 83.6% for SVJedi, cuteSV, Sniffles2, and LRcaller, respectively), as well as on both deletion and insertion subsets (Table 2). The higher genotyping accuracy of deletions over insertions observed with SVJedi-graph is also observed with cuteSV, LRcaller and Sniffles2, at an even more pronounced level, from 3.7 point % difference with cuteSV to 10 with Sniffles2.

SVJedi genotypes a lower proportion of deletions than insertions (85.8% against 93.6%), while all other four genotypers show a relatively stable genotyping rate between both SV types (at most 0.8 point % difference). It is to be noted that cuteSV and LRcaller seem to systematically assign a genotype to all input SVs, thus having a fixed genotyping rate of 100% whatever the SV set.

We also assessed the genotyping performances on the same SV set with two other long-read datasets, one of PacBio CCS (HiFi) technology, and one of ONT technology. The results obtained with the five genotypers are presented in Table 3, along with those previously obtained with the PacBio CLR dataset. SVJedi-graph shows similar genotyping performances for both PacBio datasets, whereas a small decrease in accuracy with ONT reads (of about 4 points %) with a slight increase in rate (of 0.5–1 point %). Contrary to SVJedi-graph, all other genotypers but SVJedi show a higher genotyping accuracy with HiFi and ONT reads compared to CLR reads, Sniffles2 and LRcaller having their best genotyping accuracy on this SV set with the HiFi reads (89.4% and 86.2%, respectively), and cuteSV having its best genotyping accuracy with the ONT reads (92.7%).

3.4 Applying SVJedi-graph on challenging SVs

As the High confidence set contains only distant SVs, we wanted to explore our method's performances on a more challenging SV set and we applied SVJedi-graph on another SV set from the HG002 GIAB callsets, called "ClusteredCalls" (that are included in the more difficult Tier 2 regions). This set contains 7003 SV calls that were not included in the High confidence set due to a characterization of lower quality (on breakpoint position and/or genotype). As a matter of fact, 99.5% of these SV calls are within 1 kb of at least one other call. Notably, 58% of deletions overlap at least one other deletion of the set. Additionally, 83% of these SVs fall in regions of Tandem Repeats greater than 100 bp. As the genotypes indicated in the set may not be fully considered as ground truth, we will refer to the genotype quality in terms of % of identical genotypes instead of genotyping accuracy for this set.

All genotyping tools show difficulties to genotype this SV set in comparison to the High confidence SV set, with a decrease of about 20 points of the % of identical genotypes, resulting in around 61–71% of identical genotypes for all tools (Table 4).

Our tool was able to assign a genotype to 81.5% of these SVs, and 69.4% of them with an identical genotype to the one indicated in the GIAB set, the highest value being 71.4% obtained by CuteSV. SVJedi-graph results on deletions and insertions are very contrasted, both on % of identical genotypes and on genotyping rate. We were able to genotype almost all deletions (95.3%) but with only 51.5% of identical genotypes, while we genotyped less insertions (74.9%) but with the highest % of identical genotypes (80.7%) among the five tools. Both SVJedi-graph and SVJedi showed better performances on insertions than deletions, contrary to the other three SV callers that have in common to rely primarily on read mapping on the reference genome only. SVJedi showed an impaired genotyping rate of 25.5%, which was to be expected considering its difficulties to assign genotypes in the context of close and overlapping SVs.

As concerns SVJedi-graph results, the regions with higher densities of SVs and overlapping SVs did not harbor more missing or different genotypes as in the GIAB set. We could not find any association or relationship between missing and error genotypes with SV size or Tandem Repeat context (as was the case for SVJedi in their publication; Lecompte et al. 2020) to explain the lower concordance of genotypes.

3.5 Genotyping a real not curated callset

In addition to the GIAB benchmark SV sets, we tested our method on "raw" SV calling results, which are more likely to contain nested SVs and false positive calls. The idea was to verify that the presence of "noisy" calls (either false positives or poorly described SVs) did not disrupt the genotyping quality of nearby true positive SVs. We applied SVJedi-graph on an SV callset obtained by running a single SV discovery tool, Sniffles (Sedlazeck et al. 2018), on PacBio CLR reads data from HG002 individual, containing 17 637 discovered SVs, with 7921 deletions and 9517 insertions. In this uncurated callset, 13% of the calls are <1 kb apart from another call and 2.3% of the deletions overlap at least one other deletion. SVJedi-graph attributed a genotype to 98% of the 17 637 discovered SVs. To assess the accuracy of these genotypes, we compared these SVs with the ones from the High confidence GIAB HG002 callset, by merging the two sets with Jasmine (Kirsche et al. 2021). Among the 9729 insertions and deletions identified as common between the two sets, 96.4% showed identical genotypes. This is similar and even higher

Table 3. Genotyping accuracy and rate of SVJedi-graph and state of the art genotyping tools on the deletions and insertions of the *High confidence* HG002 SV set, genotyped with PacBio CLR (30×), PacBio CCS (HiFi, 25×), and ONT (40×) reads.

	PacBio CLR		PacBio HiFi		ONT	
Tool	Accuracy (%)	Rate (%)	Accuracy (%)	Rate (%)	Accuracy (%)	Rate (%)
SVJedi-graph	94.6	99.0	94.1	99.5	90.4	100.0
cuteSV	88.7	100.0	91.3	100.0	92.7	100.0
LRcaller	83.6	100.0	86.2	100.0	84.8	100.0
Sniffles2	85.4	99.5	89.4	99.2	88.9	99.8
SVJedi	92.2	90.2	81.3	84.4	90.7	86.2

Table 4. Genotyping rate and % of identical genotypes of SVJedi-graph and state of the art genotyping tools on the deletions and insertions of the *ClusteredCalls* HG002 SV set.[a]

	Global		Deletions		Insertions	
Tool	% of identical genotypes	Rate (%)	% of identical genotypes	Rate (%)	% of identical genotypes	Rate (%)
SVJedi-graph	69.4	81.5	51.5	95.3	80.7	74.9
cuteSV	71.4	100	76.5	100	67.9	100
LRcaller	66.4	100	71.2	100	63.1	100
Sniffles2	61.1	99.7	62.4	100	60.2	99.6
SVJedi	70.3	25.5	47.7	16.9	78.5	31.2

[a] Genotyping was performed with the 30× CLR PacBio read dataset.

than the accuracy obtained on the curated high confidence set and this indicates that the presence of noise in the SV set does not prevent SVJedi-graph to accurately genotype true positive calls.

Interestingly, common SVs between the two sets did not share exactly the same breakpoint positions, with 57% of them differing by more than 10 bp and 14% by more than 50 bp. This confirms that small imprecision on the breakpoint definition does not impair SVJedi-graph genotyping quality.

3.6 Running time and memory usage

The running time and memory requirement of SVJedi-graph and the other genotypers compared on the GIAB HG002 *High confidence* SV set are shown in Table 5. When including the mapping time, SVJedi-graph took less than half an hour to genotype the whole High confidence HG002 SV callset with 30× PacBio CLR reads. It is more than six times faster than all other long-read genotypers (including the mapping time). Notably, the total SVJedi-graph time is similar to the genotyping time alone, for tools that require a mapping to the reference genome (CuteSV, Sniffles2, LRcaller), considering this file may have been obtained previously for other purposes. In terms of memory requirements, SVJedi-graph was in a similar order of magnitude than SVJedi and the two are the less memory demanding tools of the five tested, while LRcaller and Sniffles2 required about 1.5–2 times more memory, and cuteSV about three times more. For LRcaller and Sniffles2, the most memory demanding step among the whole genotyping process was the read mapping with minimap2.

For all genotypers, the most time-requiring step is the long-read mapping on the reference genome or on the variation graph for our method. The speed-up of our method is explained by the fact that we chose to use minigraph in its fastest mode, which outputs only alignment coordinates computed over the chaining of minimizers without aligning all bases in between. We also assessed the performances of our method using base-level alignments obtained with minigraph (option "-c") and another long-read mapper on graph, GraphAligner (Rautiainen and Marschall 2020). Our tests showed that using base level alignments did not improve genotyping rate and accuracy, while drastically increasing the mapping time by at least 15 times.

Table 5. Running time and memory requirements on the HG002 *High confidence* SV set.[a]

Tool	Running time (min)			Memory (Go)
	Total	Mapping	Genotyping	
SVJedi-graph	29.7	24.8	4.3	19.1
cuteSV	201.9	176	25.9	65.2 (cuteSV)
LRcaller	196.6	176	20.6	29.2 (minimap2)
Sniffles2	233.9	176	57.9	29.2 (minimap2)
SVJedi	189.9	181.9	7.5	13.9

[a] All tools were run on 20 CPU threads when multi-threading was supported (all but Sniffles2). The total running time shown for SVJedi-graph and SVJedi includes the SV representation step (allelic linear sequences for SVJedi and variation graph for SVJedi-graph) in addition to the mapping and genotyping time. The memory requirement shown for cuteSV, Sniffles2 and LRcaller is the maximum amount of memory used by either the genotyper or minimap2.

4 Discussion and conclusion

We have presented here the first method and its implementation dedicated to SV genotyping with long reads that is based on a variation graph. The use of a variation graph allows to represent in a single data structure the whole genome along with all described alternative SV alleles. In such a graph, reference and alternative alleles are represented in a strictly equal manner, preventing a potential bias toward the reference allele when mapping reads on it. We have shown on simulated deletion datasets that this approach achieves highly accurate genotyping and the few observed genotyping errors were balanced over both alleles. When applied on a simulated dataset with random inversions, we observed similarly a very high genotyping accuracy without any reference bias (see Supplementary Table S1). Further evidence of the absence of

reference bias in SVjedi-graph is the fact that it obtained similar performances between insertions and deletions in the real human benchmark SV set, in contrast to LRcaller and Sniffles.

The second major advantage of using a variation graph is that it allows to represent close and even overlapping SVs efficiently. In particular, for closely located SVs, this representation does not require to choose some haplotypes over all the possible ones. We designed simulated datasets where we controlled the distance or overlap between consecutive simulated SVs in order to precisely assess the impact of such SV distributions on the genotyping performances of the tools. In these simulations, we did not modify the simulated haplotypes and resulting simulated sequencing reads, but we only added additional SVs in the input SV set. The latter are thus to be genotyped as homozygous for the reference allele (0/0). This is the simplest case of close or overlapping variants, since the genotyping signals contained in the reads should remain the same whatever the additional set of SVs. Even in this simplest case, we observed a substantial decrease in genotyping rate or accuracy for all tools except SVJedi-graph as soon as SVs are $<$500 bp apart or overlapping. It means that in these methods, the quality of the genotyping of a given SV depends on the other SVs present in the SV set, even if absent in the genotyped individual. This was to be expected for SVJedi as it constructs linear allelic sequences around each SV breakpoint independently of the other SVs in the set. As these sequences span up to 5 kb on either side, when the SVs are close, the resulting set of sequences has a lot of redundancy, causing many reads to be filtered out due to their non-unique mapping. This explains why the genotyping rate of SVJedi drops drastically in these results. On the contrary, the stable performances of SVJedi-graph on these datasets demonstrates that the graph-based representation of SVs prevents such non desirable behavior, and allows highly accurate genotyping of clustered and even overlapping SVs.

The real HG002 High confidence benchmark dataset from Genome in a Bottle consortium does not contain such clustered or overlapping SVs, since all SV calls have been selected to be at least 1 kb apart from one another in order to ensure this high confidence in the SV descriptions and genotypes. However, it still contains challenging insertions and deletions, since for instance more than half of them are contained in Tandem Repeat regions greater than 100 bp (Zook et al. 2020, Delage et al. 2020). On this dataset dedicated to the evaluation of SV tools, SVJedi-graph obtained substantial improvement in genotyping accuracy and rate with PacBio CLR and HiFi reads compared to the other tested genotypers and in much less time. Although most other tools had better genotyping performances with ONT reads on this dataset, SVJedi-graph showed a lower genotyping accuracy with respect to the ones obtained with PacBio sequencing reads. This may be explained by the fact that the mapping in SVJedi-graph was performed with the same default parameters of minigraph for all sequencing technologies, whereas the mapping used by other tools was performed with sequencing technology specific parameter presets of minimap2. For the moment, minigraph does not provide parameter presets for the different sequencing technologies, an exploration of mapping parameters that would be best suited to the different technologies could lead to improvements in SVjedi-graph.

On a more challenging SV set with many clustered and overlapping calls, we could have expected based on the simulation results that SVJedi-graph would make an even greater difference with other tools. On the contrary, we obtained poor concordance with the genotypes given as the truth in the HG002 ClusteredCalls set of GIAB, with similar or sometimes worse values than the other genotypers. We investigated numerous factors to explain these results, including the proximity or overlapping of SVs, the genomic context of SVs, the size or genotypes of erroneously genotyped SVs but we did not find any significant association. This absence of relationship with classical factors of errors may argue toward problems of definition of SV breakpoints or inacurrate genotypes in the input SV set. Indeed, the authors of this dataset had deliberately distinguished them from the High confidence set and had warned users that the proximity of the SVs prevented them from being accurately characterized and that they were "potentially complex, compound, or inaccurate" (citation from the repository Readme). The fact that some of the tested genotypers perform better on this particular dataset could be due to biases or errors that are reproducible with similar methods. Indeed, CuteSV and Sniffles genotypings are derived from discovery methods and rely on the same input data, namely reads mapped on the reference genome. They probably use similar read signals that were used to discover these SVs in the first place in GIAB protocols. For instance, they may have used the variation of read depth along the genome to discover some deletions, whereas SVJedi-graph relies exclusively on the breakpoint signals. Notably, those methods performed worse for insertions, for which the signals that can be extracted from mapping to the reference genome are the weakest. In the case of insertions, we notably observed that SVJedi-graph had the best genotyping accuracy but was not able to genotype more than 25% of them due to insufficient read support (less than three reads) for both alleles combined. Interestingly, 85% of these not genotyped insertions are reported as homozygous for the alternative allele in the input set. Such absence of read support even for the reference allele could be explained by inaccuracies in the reported inserted sequence which would be too divergent from the real insertion sequence for reads to map on. These different hypotheses are difficult to settle other than by a manual inspection of each individual case, which would be extremely time-consuming and is outside the scope of this paper.

The uncertainties in this dataset make it therefore poorly suited for precise assessment of tool performances. Conversely, while the High confidence set is an ideal set for benchmarking and comparing tools, it does not reflect the reality of genotyped datasets in practice, which are usually not manually curated and contain more closely located and nested calls, as well as more imprecise and noisy calls. Our experiment on a whole raw and uncurated discovery call set represents a practical and realistic intermediate between the high confidence and the most challenging call sets and showed that SVJedi-graph is usable and obtains good quality results in practice in a few dozens of minutes on a whole human genome dataset.

In conclusion, SVJedi-graph is a fast and efficient tool to genotype SVs with long-read data, that promises to be useful in the ever-growing number of population-scale SV studies.

Acknowledgements

We acknowledge the GenOuest bioinformatics core facility for providing the computing infrastructure. We thank Lolita

Lecompte for providing some of the scripts to generate simulated datasets. We are also grateful to the anonymous reviewers for their insightful and constructive comments.

Supplementary data

Supplementary data is available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This project has received funding from the French Agence Nationale de la Recherche ANR-20-CE02-0017 Divalps grant. A CC-BY public copyright license has been applied by the authors to the present document, in accordance with the grant's open access conditions.

Data availability

The data underlying this article are available on public repositories whose links are given in the article online supplementary material.

References

Alonge M, Wang X, Benoit M *et al.* Major impacts of widespread structural variation on gene expression and crop improvement in tomato. *Cell* 2020;**182**:145–61.e23.

Beyter D, Ingimundardottir H, Oddsson A *et al.* Long-read sequencing of 3,622 icelanders provides insight into the role of structural variants in human diseases and other traits. *Nat Genet* 2021;**53**:779–86.

Chaisson MJP, Sanders AD, Zhao X *et al.* Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun* 2019;**10**:1784.

Chen S, Krusche P, Dolzhenko E *et al.* Paragraph: a graph-based structural variant genotyper for short-read sequence data. *Genome Biol* 2019;**20**:291.

Coster WD, Weissensteiner MH, Sedlazeck FJ. Towards population-scale long-read sequencing. *Nat Rev Genet* 2021;**29**:572–87.

Delage WJ, Thevenon J, Lemaitre C *et al.* Towards a better understanding of the low recall of insertion variants with short-read based variant callers. *BMC Genomics* 2020;**21**:762.

Duan X, Pan M, Fan S *et al.* Comprehensive evaluation of structural variant genotyping methods based on long-read sequencing data. *BMC Genomics* 2022;**23**:324.

Ebler J, Ebert P, Clarke WE *et al.* Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. *Nat Genet* 2022;**54**:518–25.

Eggertsson HP, Jonsson H, Kristmundsdottir S *et al.* Graphtyper enables population-scale genotyping using pangenome graphs. *Nat Genet* 2017;**49**:1654–60.

Eggertsson HP, Kristmundsdottir S, Beyter D *et al.* GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nat Commun* 2019;**10**:5402.

Garrison E, Sirén J, Novak AM *et al.* Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat Biotechnol* 2018;**36**:875–9.

Guarracino A, Heumos S, Nahnsen S *et al.* ODGI: understanding pangenome graphs. *Bioinformatics* 2022;**38**:3319–26.

Jiang T, Liu Y, Jiang Y *et al.* Long-read-based human genomic structural variation detection with cuteSV. *Genome Biol* 2020;**21**.

Kirsche M, Prabhu G, Sherman R *et al.* Jasmine and Iris: population-scale structural variant comparison and analysis. *Nat Meth* 2023;**20**:408–17.

Lecompte L, Peterlongo P, Lavenier D *et al.* SVJedi: genotyping structural variations with long reads. *Bioinformatics* 2020;**36**:4568–75.

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

Li H, Feng X, Chu C *et al.* The design and construction of reference pangenome graphs with minigraph. *Genome Biol* 2020;**21**

Mahmoud M, Gobet N, Cruz-Dávalos DI *et al.* Structural variant calling: the long and the short of it. *Genome Biol* 2019;**20**:246.

Nielsen R, Paul JS, Albrechtsen A *et al.* Genotype and SNP calling from next-generation sequencing data. *Nat Rev Genet* 2011;**12**:443–51.

O'Donnell S, Yue JX, Saada OA *et al.* 142 telomere-to-telomere assemblies reveal the genome structural landscape in *Saccharomyces cerevisiae*. bioRxiv 2022. https://doi.org/10.1101/2022.10.04.510633.

Paten B, Novak AM, Eizenga JM *et al.* Genome graphs and the evolution of genome inference. *Genome Res* 2017;**27**:665–76.

Phan L, Hsu J, Tri LQM *et al.* dbVar structural variant cluster set for data analysis and variant comparison. *F1000Res* 2017;**5**:673.

Porubsky D, Höps W, Ashraf H *et al.* Recurrent inversion polymorphisms in humans associate with genetic instability and genomic disorders. *Cell* 2022;**185**:1986–2005.e26.

Rautiainen M, Marschall T. GraphAligner: rapid and versatile sequence-to-graph alignment. *Genome Biol* 2020;**21**:253.

Sedlazeck FJ, Rescheneder P, Smolka M *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat Methods* 2018;**15**:461–8.

Sirén J, Monlong J, Chang X *et al.* Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science* 2021;**374**:abg8871.

Stöcker BK, Köster J, Rahmann S *et al.* SimLoRD: simulation of long read data. *Bioinformatics* 2016;**32**:2704–6.

Weischenfeldt J, Symmons O, Spitz F *et al.* Phenotypic impact of genomic structural variation: insights from and for human disease. *Nat Rev Genet* 2013;**14**:125–38.

Zhao X, Weber AM, Mills RE *et al.* A recurrence-based approach for validating structural variation using long-read sequencing technology. *Gigascience* 2017;**6**:1–9.

Zook JM, Hansen NF, Olson ND *et al.* A robust benchmark for detection of germline large deletions and insertions. *Nat Biotechnol* 2020;**38**:1347–55.

IV Large Inversion Discovery In the Alpine Coenonympha Complex of Species

In this chapter

1 Introduction 47
Paper: Characterization of large inversions to investigate hybrid speciation in the four species-complex of alpine *Coenonympha* butterfly 48

1 Introduction

One of the goals of the DIVALPS project is to explore the genetic and genomic factors involved in the evolutionary history of the complex of alpine *Coenonympha* species. To this end, genome sequencing data for the four *Coenonympha* species were produced, including long PacBio CLR reads for one individual per species and short Illumina reads for 9 to 19 individuals per species. The long read sequencing data led to the production and publication of the first assembled genome for *Coenonympha arcania* (Legeai et al., 2024). I contributed to this work through the quality evaluation of the produced genome (k-mer completeness and multiplicity), its comparison to the reference genomes of three close species (synteny analysis) and in the search for the unannotated W sex chromosome (long read mapping). These data also led to the production of the first assembled genomes for the three other species, which are presented in this chapter. These high quality genomes were critical to this work, in which I compare the four *Coenonympha* genomes. I tested many different tools to discover large inversions between these genomes, from SV discovery tools based on long-read mapping or assembly-alignment, more fitted for intra-species comparison, to genome alignment tools made for inter-species comparison. In this paper, we focused on the large inversions discovered by the assembly-based SV discovery method that performed the best on the *Coenonympha* genomes.

The frequent presence of repeated patterns near inversions extremities, as well as the sequence divergence between genomes, have made large inversions between species challenging to precisely characterize from short or long read alignment. New methods based on whole-genome alignment were shown to overcome these difficulties. Using such method, I discovered 12 large ($\geq$ 100 kbp) inversions among the four species, of which 2 were detected in the heterozygous state in the assembled genome of *C. cephalidarwinianna*. Complementary analyses of the population genetic data statistics in the inversions, which were obtained from the work of our collaborator in the project Capblancq *et al.* (*in prep*), suggest that at least 3 inversions can be found as heterozygous in populations of *C. cephalidarwinianna*, and that 5 harbour barriers to gene flow.

This chapter is written in the form of a paper, which is in its way to be submitted to a journal of genome biology and evolutionary genomics in the upcoming months. I produced all the results presented in this paper, at the exception of the genomme assemblies, the gene annotations and the population genetic data statistics.

Characterization of large inversions to investigate hybrid speciation in the four species-complex of alpine Coenonympha butterfly

Sandra Romain[1], Thibaut Capblancq[2], Laurence Després[2], Mathieu Joron[3], Fabrice Legeai[1,4], and Claire Lemaitre[1]

[1] *Inria, CNRS, IRISA, University of Rennes, 35000 Rennes, France*
[2] *LECA, CNRS, Université Grenoble-Alpes, Université Savoie Mont Blanc, Grenoble, France*
[3] *CEFE, CNRS, EPHE, IRD, Université de Montpellier, Montpellier, France*
[4] *IGEPP, INRAE, Institut Agro, University of Rennes, 35653 Le Rheu, France*

1 Introduction

Speciation is a fundamental process responsible for the diversity of life, yet the genomic bases of speciation remain poorly understood. The most striking examples of speciation include cases where lineages diverge without geographical isolation, as they can still exchange genes, referred to as speciation with gene flow (Nosil, 2008; Smadja and Butlin, 2011; Feder et al., 2012). Speciation with gene flow (or sympatric speciation) has long been thought as highly unlikely because gene flow counteracts the effect of divergent selection but is now more and more often documented thanks to the access to whole genome polymorphisms and evidence of heterogenous gene flow along the genome. Indeed, some genes may still be exchanged across lineages while other included in so-called 'barrier loci' are not. These barrier loci can be involved in genetic incompatibilities (the hybrid zygote is non viable, or non fertile), or involved in ecological specialisation where the hybrids are maladapted to either parental environments (post-zygotic isolation) and/or in mate choice (prezygotic isolation). As a result, this may lead to ecological speciation, where the two diverging lineages are adapted to different habitats, and hybrid speciation, where a new species originates from the combination of two parental genomes. In the face of gene flow, large inverted non-recombining streches of DNA provides a way for clusters of adaptive genes or genes involved in reproductive isolation to avoid recombination with sister or parental lineages (Jay et al., 2018; Joron et al., 2011). Indeed, when inversions are in the heterozygous state in a diploid genome (i.e. presence of both the ancestral and inverted alleles), a single crossover during meiosis within the inversion generates unbalanced gametes that contain duplication and/or deletion. As a result, the effective rate of recombination in such region is highly reduced, and these inversions can result in advantageous allelic combinations to remain in full association (Yeaman, 2013; Kirkpatrick, 2010; Huang and Rieseberg, 2020; Berdan et al., 2023).

The role of inversions in promoting genome divergence and speciation is a growing and dynamic field in evolutionary biology research and in the past decade, numerous studies have reported inversions associated with adaptative phenotypes, behaviour, mating strategies and speciation across

various clades (Wellenreuther et al., 2019; Wellenreuther and Bernatchez, 2018; Gabur et al., 2019; Weischenfeldt et al., 2013). For example, the reduced recombination rate between the two alleles of an inversion can promote adaptation by maintaining advantageous allele combinations, or even cause an accumulation of mutations leading to reproductive isolation (Berdan et al., 2023). An extreme example of phenotypic polymorphism related to inversions capturing specific allele combinations is supergenes, which capture multiple phenotypic characters while incuring balanced polymorphism in the populations (Thompson and Jiggins, 2014). A great example of supergenes is the P supergene of the Heliconius numata butterfly which contains three large inversions and is involved in locally-adaptated wing pattern polymorphism (Joron et al., 2011).

But inversions are difficult to accurately characterize in genomes directly from short and long read alignments (Liu et al., 2024). Even more so for large inversions, due to the frequent presence of repeated patterns near inversion borders (Sanders et al., 2016) that hinders read alignment and makes population genotyping difficult. Moreover, by impeding proper read mapping, increased sequence divergence between species makes it more challenging to precisely characterize such variants than within species. But, with the release of high quality, ultra-long and Hi-C reads and the development of new software, de-novo chromosome-level assembly is more accessible for various diploid genomes (Li and Durbin, 2024). The higher accuracy of contigs compared to long or short reads allows for more precise inference of structural variants and their greater length improves the discovery of large-scale rearrangements directly from whole genome alignments (WGA) with dedicated tools (Nattestad and Schatz, 2016; Goel et al., 2019; Heller and Vingron, 2020; O'Donnell and Fischer, 2020), as observed for different cultivated plant systems (Zhou et al., 2023; Li et al., 2023).

The evolutionary history of the alpine species complex of Coenonympha butterflies makes them a good model to study the genomic factors of speciation and adaptation in this genus. Genomic studies suggest that the two species *Coenonympha darwiniana* and *Coenonympha cephalidarwiniana* originated from a unique ancestral hybrid population following the hybridization of *Coenonympha arcania* and *Coenonympha gardetta* during the last glacial cycle ∼10,000–20,000 years ago (Capblancq et al., 2015, 2019). The relative contribution of the two parental genomes to the ancestral hybrid species was estimated to be strongly asymetrical with about 75% inherited from arcania and 25% from gardetta. The two parental species occupy distinct ecological niches, with *C. arcania* being distributed at altitude below 1,500 m, and *C. gardetta* occupying higher altitude range over 1,500 m. The two hybrid lineages are found between 1,300 and 2,500 m, and occasionally they coexist with their parental species in narrow zones (Capblancq et al., 2019). Gene flow in the species complex is not homogeneously distributed along the genome. Between the parental species, many genomics regions were identified as strong barriers to gene flow, covering about 6% of the genome and mostly located on the sex Z chromosome (Capblancq et al, submitted). Little is known about the factors that drove and now maintain differentiation among these four species, especially at the genomic scale.

In this study, we identified large genomic inversions that could have played a role in the diversification of these butterflies, and help the two hybrid lineages to isolate from their parental species. To this end, we produced chromosome level genome assemblies for three new Coenonympha species (*C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana*), and compared them to a previously published genome of *C. arcania* (Legeai et al., 2024) to identify large genomic inversions present in the complex using a whole-genome alignment approach. Among the 12 identified large inversions, we found at least four inversions with patterns of population diversity contrasting with the rest of the genome, and including genes potentially related to adaptation.

2 Results

2.1 Assembly statistics and gene annotation

In addition to the genome of *Coenonympha arcania*, already published in Legeai et al. (2024), we sequenced three new genomes from the species complex in PacBio HiFi and assembled them with Hifiasm (Cheng et al., 2022) with an extra step of purge_dups (Guan et al., 2020) in order to remove haplotypic duplications, following the exact same procedure as for the first genome. Although more fragmented than the *C.arcania* genome, which had benefited from complementary Omni-C sequencing, the produced assemblies are close to the chromosomal level, allowing the detection of inversions (table 1). We annotated protein-coding genes from the 4 genomes in a similar way using Helixer (Holst et al., 2023), here again following the procedure conducted for *C. arcania* reference genome annotation. Although Helixer does not require transcriptomic data, the number of genes identified in the new three genomes is similar to the one identified in the *C. arcania* initial reference and BUSCO scores are good (table 1).

Table 1: Assembly and annotation metrics of the 4 *Coenonympha* genomes. Read depth was calculated with a 500 Mbp genome size.

Metrics	*C.arcania*	*C. gardetta*	*C. darwiniana*	*C. cephalidarwiniana*
read number	1,413,972	1,132,959	1,823,533	1,403,689
read median length (bp)	13,587	12,115	11,866	12,032
read mean length (bp)	13,963	12,787	12,658	12,728
estimated read depth	39.5	29.0	46.2	35.8
contig number	38	99	55	50
genome size (Mbp)	497.3	523.2	487.0	477.9
genome N50 (Mbp)	17.9	17.2	17.7	17.2
genome BUSCO single (%)	97.3	97.1	96.8	95.8
genome BUSCO duplicated (%)	0.9	1.1	1.5	1.1
genome BUSCO fragmented (%)	0.4	0.5	0.4	0.4
genome BUSCO missing (%)	1.4	1.3	1.3	2.7
gene number	21,392	21,347	20,499	20,492
annotation BUSCO single (%)	94.0	93.5	93.6	91.9
annotation BUSCO duplicated (%)	1.7	2.3	2.4	2.1
annotation BUSCO fragmented (%)	1.4	1.4	1.4	1.8
annotation BUSCO missing (%)	2.9	2.8	2.6	4.2

2.2 Twelve large curated inversions across the complex

To identify large inversions between the four Coenonympha genomes, we used SyRI (Goel et al., 2019), which detects structural variants from a pairwise whole genome alignment achieved with minimap2 (Li, 2018). In order to provide a unique coordinate referential and ease the merging of variants between the sets, we used *C. arcania* as the reference genome against which we compared the other three genomes of the complex (*C. gardetta*, *C. darwiniana*, *C. cephalidarwiniana*) and another Coenonympha species: the chestnut heath C. glycerion (ENA project PRJEB71111). The

Table 2: Location and size of the twelve inversions of the curated callset, with reference to *C. arcania* genome.

Chromosome	Start position (bp)	End position (bp)	Size (kb)	Identifier
Z	162,190	4,653,773	5,492	Z.1
	3,819,827	4,373,994	100	Z.2
	16,680,467	19,193,380	2,513	Z.3
6	8,530,155	9,001,193	471	6.1
	11,827,505	13,547,418	1,720	6.2
10	4,320,561	5,459,813	1,139	10.1
	7,372,573	7,478,285	106	10.2
14	12,069,599	12,513,289	444	14.1
21	6,611,667	7,279,214	668	21.1
26	2,842,557	3,243,102	401	26.1
	3,305,509	3,855,539	550	26.2
28	7,162,273	7,492,976	331	28.1

C. glycerion genome was added to the comparison, as an outgroup species, to provide insight into the history of the inversions detected in the complex, and so inversions specific to *C. glycerion* were disregarded. After filtering out those of size smaller than 100 kb, we obtained 14, 13 and 12 large inversions on *C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana*, respectively.

After merging the four sets and a manual inspection of each inversion (see Material and Methods section), we obtained a total of 12 curated large inversions within the four species-complex, of sizes spanning 100 kb to 5 Mb (Tab.2, Fig.1). Three inversions are located on the sex chromosome Z, and nine are located on autosomes: six on the chromosomes 6, 10 and 26, and three on chromosomes 14, 21 and 28.

2.3 Four presence-absence patterns for the inversions

The merged set of validated inversions shows four different distribution patterns across the *Coenonympha* complex relative to the outgroup species *C. glycerion* (Fig. 2). Of the four patterns, two depict inversions detected in at least two species, and two depict inversions detected in only one species. More than half of the inversions (8 out of 12) were detected only in *C. arcania* (on chromosomes Z, 6, 14, 21, and 26).

All inversions are found between *C. arcania* and *C. gardetta*, which is to be expected considering their more ancient time of divergence, about 1.7MYA (Capblancq et al., 2015). Among these inversions, the structural form inherited by the hybrid species was more frequently transmitted by *C. gardetta*.

Interestingly, the inversion INV 10.1 identified in only one of the parent (i.e *C. arcania*) was inherited differently between the two hybrid species.

Three autosomal inversions were detected as heterozygous in at least one of the sequenced individuals. Inversions 6.1 and 21.1 were found heterozygous in *C. cephalidarwiniana*, and inversion

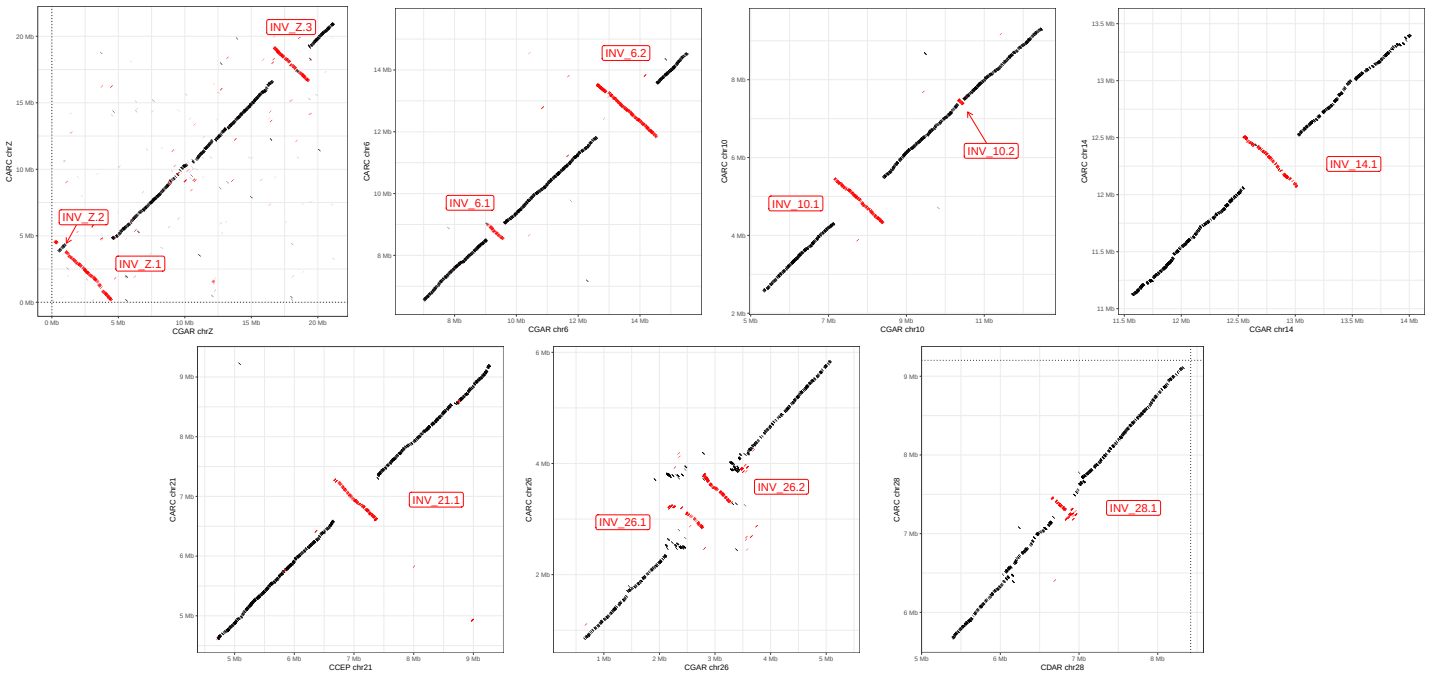

Figure 1: Local dotplots of pairwise alignments of the seven chromosomes bearing inversions. For all dotplots, Y-axis represents the position of the alignments on *C. arcania* reference chromosomes. X-axis represents the position of the alignments on *C. gardetta* for chromosomes Z, 6, 10, 14 and 26, on *C. cephalidarwiniana* for chromosome 21, and on *C. darwiniana* for chromosome 28, which were the genome pairs with the clearer display of each inversion. Black lines represent forward alignments, red lines represent reverse alignments.

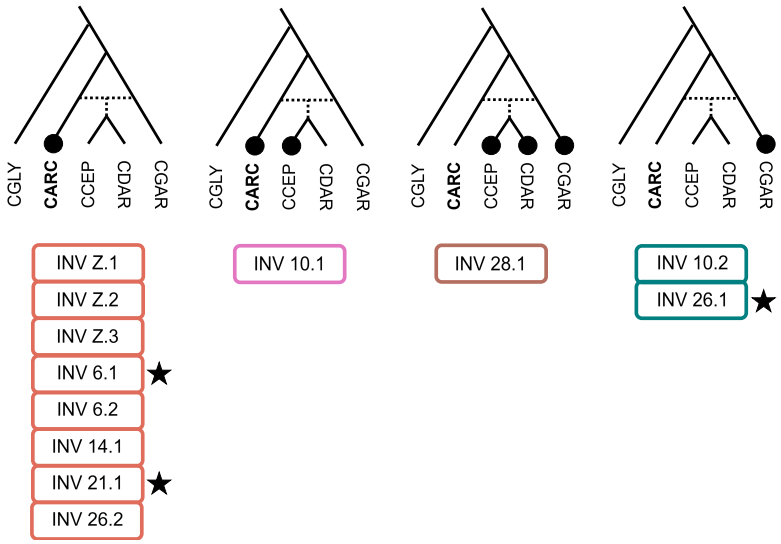

Figure 2: Patterns of the large inversions detected across the *Coenonympha* species tree (as inferred from Capblancq et al. (2019)), including the outgroup species *C. glycerion*. The observed inversions relative to *C. glygerion* genome are represented in black dots. The boxes under the trees contain the identifiers of the inversions in each distribution pattern. The black stars indicate inversions for which heterozygosity was detected in at least one species. CGLY: *C. glycerion*, CARC: *C. arcania*, CCEP: *C. cephalidarwiniana*, CDAR: *C. darwiniana*, CGAR: *C. gardetta*.

26.1 was found heterozygous in *C. gardetta*.

2.4 Population genetics statistics inside inversions

Furthermore, we investigated if the inverted loci contained in these 12 inversions showed contrasted pattern of evolution by measuring locally the genetic diversity within various populations resequenced for another study (Capblancq *et al.*, in prep.). The variant calling dataset was obtained from the mapping of short reads along the *C. arcania* genome of 19, 17, 16 and 9 individuals of respectively *C. arcania*, *C. gardetta*, *C. cephalidarwiniana* and *C. darwiniana*.

We first questioned whether the inversions contained *loci* identified as impermeable to gene flow between the parental species and called hereafter *genomic barriers*. These loci cover about 6% of the genome and are mostly located on the sex Z chromosome (Capblancq *et al.*, in prep.). Consistently, all three inversions located on the Z chromosome show more than 60 % of genomic barriers. Notably, we found that the autosomal inversions 6.1 and 21.1, both previously reported as heterozygous in *C. cephalidarwiniana*, are nearly entirely composed of genomic barriers (95 to 100 % of their locus length), contrasting with the other inversions and other autosomal *loci* being almost all bare of such barriers (Fig. 3).

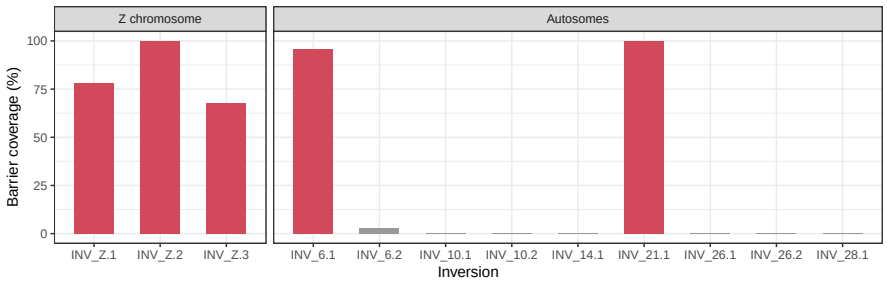

Figure 3: Percentage of the 12 large inversion loci covered by signals of genomic barriers (impermeable to gene flow) between *C. arcania* and *C. gardetta*. The signals were detected on windows of 50 kb.

While the majority of the autosomal genome of the hybrid species was mainly inherited from *C. arcania*, the genomic content of 5 and 3 inversions was most likely inherited from *C. gardetta* in *C. darwiniana* and *C. cephalidarwiniana*, respectively (inversions 6.1, 10.1, 10.2, 14.1, 21.1 ; and inversions 6.1, 10.2 and 14.1) (Fig. 4). Such feature could originate from introgression of inversions genomic content from *C. gardetta*, and raises the question as to whether these inversions propagated across *C. darwiniana* and *C. cephalidarwiniana* populations under the effect of neutral evolution or directional selection. The inversions 10.1 and 21.1 show differing ancestries for *C. darwiniana* and *C. cephalidarwiniana*, which is consistent with the species tree pattern (Fig. 2) for the inversion 10.1, but not for the inversion 21.1. On the Z chromosome, which, unlike the autosomes, has been predominantly inherited from *C. gardetta* in the hybrid species (Capblancq *et al.*, in prep.), the inversions also appear to be of *C. gardetta* ancestry.

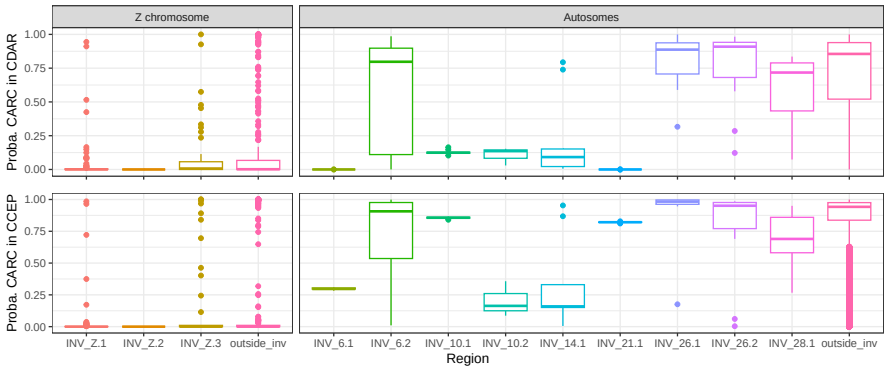

Figure 4: Probability of assignation to *C. arcania* population from admixture on SNP for regions in *C. darwiniana* genome ('Proba. CARC in CDAR', top) and *C. cephalidarwiniana* genome ('Proba. CARC in CCEP', bottom). The boxplots represent the distribution of assignation probability for 50 kb windows inside each inversion region ('INV_*') and for non-inversion regions ('outside_inv') in the Z chromosome (left panel) or across the autosomes (right panel).

We explored the four highlighted autosomal inversions 6.1, 10.1, 14.1 and 21.1, in order to identify local patterns of population diversity with various genetic statistics (Fig. 5) calculated in 50kb windows along the *C. arcania* genome. We did not include inversion 10.2 as its size of $\sim$ 100 kb did not provide enough usable/informative datapoints. Figure 5 shows that the genetic barriers found in inversions 6.1 and 21.1 are almost exclusively contained between the inversion breakpoints, hinting towards those inversions being the barrier themselves. It also shows that the switch between *C. arcania* and *C. gardetta* ancestry happens right at the inversion breakpoints in *C. darwiniana* and *C. cephalidarwiniana*. The local PCA analysis used to decipher different population structures all along the genome (Li and Ralph, 2018) shows a very heterogeneous signal for the inversions 6.1, 10.1 and 21.1, with three fully disassociated SNP distributions throughout the inversion loci, one for each parental species that could reflect homozygosity, and an intermediate one that could reflect heterozygosity. Interestingly, all three SNP distributions for these three inversions can be found in *C. cephalidarwiniana*, and the corresponding local admixture confirms that these inversions are not fixed in this hybrid species.

As for the genetic diversity, we noticed that inversion 10.1 shows a decrease of Tajima's D for *C. gardetta* compared to the surrounding region, with two exceptionally low points below -1. There is also peaks of Fst and linkage disequilibrium on the whole inversion 6.1 and near the end of inversion 21.1.

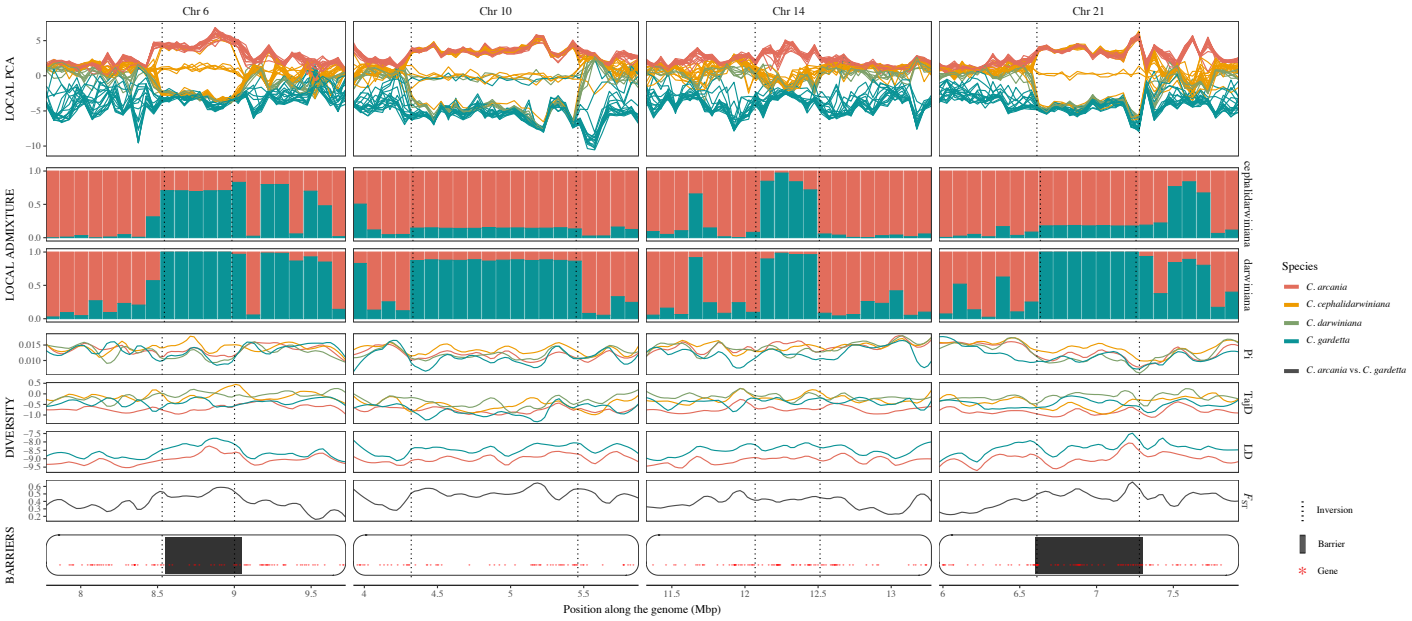

Figure 5: Population genetic statistics inferred from SNP diversity across populations of *C. arcania*, *C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana*, inside and nearing four selected large inversions (6.1, 10.1, 14.1, 21.1). All statistics were calculated on 50 kb windows, the dotted vertical lines represent inversion breakpoints position. *First row:* Local PCA of SNP distribution, each line represents one population coordinates on the first PCA dimension, with one individual per population. *Second row:* Local admixture of *C. darwiniana* and *C. cephalidarwiniana*. *Third row:* Genetic diversity estimated by π diversity (Pi) and Tajima's D (TajD) for each species, linkage disequilibrium (LD) for *C. arcania* and *C. gardetta*, and Fst between *C. arcania* and *C. gardetta*. *Fourth row:* Genomic barriers (black boxes) between *C. arcania* and *C. gardetta*, along with annotated genes (red dots) on the selected regions.

2.5 Gene functionality in inversions

Overall, 130 genes were predicted by Helixer in inversions 6.1, 10.1, 14.1 and 21.1. Of the 38 predicted genes in the inversion 10.1, one was annotated as the protein dissatisfaction gene (DSF), associated to courtship behaviors in *Drosophila melanogaster* (fruitfly) (Finley et al., 1997), and four other genes are involved in morphological development in *D. melanogaster*. Of the four, one was annotated as protein decapentaplegic gene (DPP) reported to be a key morphogen in the development during embryogenesis and postembryogenesis (Matsuda et al., 2016), one as protein abrupt-like gene reported to have a role in establishing and maintaining embryonic muscle attachments, adult sensory cell formation (macrochaetae) and morphogenesis of adult appendages (Hu et al., 1995), and two as protein SLY1 homolog (SLH) gene reported to be involved in wing disc dorsal/ventral pattern formation (Bejarano et al., 2008).

Of the 47 predicted genes in the inversion 21.1, two were annotated as sodium channel protein 60E-like gene, involved in olfactory behavior in *D. melanogaster* (Kulkarni et al., 2002). One gene

was annotated as coding for the RNA-binding protein Spenito, required for sex determination in *D. melanogaster*(Yan and Perrimon, 2015).

Of the 12 predicted genes in the inversion 6.1, three were annotated as orexin receptor type 1 (OX1R) like gene. OX1R was reported to be involved in feeding, water intake, spatial learning and reward pathways in mammals (Haynes et al., 2000; Rodgers et al., 2001; Akbari et al., 2006).

Among the 33 predicted genes in the inversion 14.1, we identified two olfactory receptors (4K2-like), the cytoskeletal protein flightless known as being involved in the regulation of many cellular processes (Strudwick and Cowin, 2020).

3 Discussion

3.1 Number of large inversions in the Coenonympha complex compared to other Satyrine

A previous study of chromosomal rearrangement in Satyrinae butterflies highlighted the recurring presence of inversions between species in this clade (Pazhenkova and Lukhtanov, 2023), where 9, 7 and 1 large inversions were detected between *Maniola jurtina* and *Erebia ligea*, between *Erebia aethiops* and *M. jurtina*, and between *E. ligea* and *E. aethiops*, respectively. However, these two genera, *Maniola* and *Erebia*, diverged more than 30MYA, far more than the estimated divergence between *C. arcania* and *C. gardetta* (1.8 MYA). Therefore, the alpine *Coenonympha* species complex seems enriched in inversions compared to the other available samples of Satyrine species. The Z chromosome seems particularly enriched, since only one inversion was detected on the Z chromosome in the *Maniola* and *Erebia* genera. Notably, this inversion was reported to be likely associated to chromosomal speciation in Satyrinae.

3.2 The contribution of parental inversions to the hybrid genomes

Although parental contribution to the ancestral hybrid population was estimated to be strongly asymetrical, with roughly 75% *arcania* and 25% *gardetta*, most of the inversions (9/12) were inherited from *gardetta* in the ancestral hybrid species, and only two originated from *arcania*. In fact, it seems that a sizeable part of the autosomal genomic content that the hybrid species inherited from *C. gardetta* is specific to inversion loci. The inversion 10.1 was putatively polymorphic in the ancestral hybrid population as it remains polymorphic in cephalidarwiniana but differently inherited from arcania to cephalidarwiniana and from gardetta to darwiniana. Furthermore although the two hybrid species originated from a single ancestral population, they appear to have retained differently this inversion.

3.3 Three autosomal inversion candidates for association to adaptation and speciation

Inversions between the parental species found differentially distributed between the hybrid species are particularly interesting, as they could hold specific gene arrangements involved in differential adaptation-linked phenotypes between the species. Combining single genome alignment and population SNP polymorphism results, we found two of such inversions on chromosomes 10 and 21, of sizes of 1.1 and 0.7 Mb, respectively. The inversion on chromosome 21 appears to be particularly interesting with signals of genetic barriers to gene flow and selection between the parental species.

The signals of genetic barriers also allowed to highlight a third autosomal inversion on chromosome 6. Interestingly, these three inversions seem to be closer to fixation in *C. darwiniana* populations than in *C. cephalidarwiniana* populations. Under the assumption that each inversion introgressed at approximately the same time in both hybrid species (in the ancestral hybrid population), this different level of fixation could rise the hypothesis of ancestral and inverted alleles each providing local adaptation advantages depending on the *C. cephalidarwiniana* population, or even be under the effect of overdominance in some populations for inversion 6.1.

The inversion on chromosome 10 appears as a good candidate for being associated with the speciation process, as it captured an assortment of genes putatively linked to both developmental of morphological features and recognition of sexual partners. Its high level of differentiation between the parental taxa and the two local drops in Tajima's D in *C. gardetta* genome hints towards influence of directional selection, which are however not observed in any of the hybrid lineages.

4 Materials and Methods

4.1 DNA Extraction and Sequencing

We used the published genomes of *C. arcania* (Legeai et al., 2024) and the assembled genome of *C. glycerion* (accession GCA_963855885.1). For the *C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana* genomes, high molecular weight DNA was extracted using Gentra Puregene kit from Qiagen, according to the manufacturer's instructions. The PacBio long-read sequencing was performed at the GenoToul Platform (Toulouse, France). Final DNA purity and concentrations were measured by spectrometry using Nanodrop and fluorometry using Qubit (ThermoFisher).

4.2 Genome assemblies and annotation

The long reads from each genome were assembled similary using Hifiasm (Cheng et al., 2022), with an extra step to remove haplotypic duplications with purge_dups v1.2.5 (Guan et al., 2020). The protein coding genes were annotated using Helixer v0.3.0 (Holst et al., 2023) with the option '–lineage invertebrate'. Functional annotation of the genes have been done with Diamond v2.0.13 (Buchfink et al., 2021) on NCBI NR 2024-6-5, Blast2GO Command Line v1.5.1 (Gotz et al., 2008) with the databases OBO=2023-07-20 and BLAST2GO=2022_08, as well as EGGNOG v2.1.9 (Huerta-Cepas et al., 2018) with database 5.0.2 and Interproscan v5.59-91.0. The completeness of the assembly and of the annotation were assessed with BUSCO v5.2.2 (Simão et al., 2015) using lepidoptera_odb10 as reference.

4.3 Identification of the inversions

The inversions between the fives genomes were characterized using SyRI (Goel et al., 2019). In order to order the scaffolds of *C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana* genomes to achieve the required chromosome-level assemblies, we first aligned their haploid assemblies, as well as *C. glycerion* genome individually against that of *C.arcania* with minimap2 (Li, 2018) v2.15 using the '-x asm20 –eqx' parameters, then we used the alignments to scaffold them with chroder (Goel et al., 2019) v1.5.4 using the default parameters. When several scaffold arrangements were possible in regions with rearrangements, we chose the arrangement resulting in the parsimonious scenario

(ie. minimizing the number of rearrangements). This happened only once, for the Z chromosome in *C. cephalidarwiniana* with one scaffold overlapping the Z.1 inversion.

We then aligned the obtained chromosome-level genomes on *C. arcania* with minimap2 using the same parameters as used for the scaffolding step. Finally, we ran SyRI (Goel et al., 2019) v1.5.4 on these alignments using the default parameters and obtained one callset per genome against the *C. arcania* reference.

We also used SVIM-asm (Heller and Vingron, 2020) haploid v1.0.2 with default parameter except –max_sv_size 5000000 and MUM&Co (O'Donnell and Fischer, 2020) v3.7 and v3.8 with default parameters but none of the tools were able to identify more than one of the large inversions observed on the alignment plots.

In order to detect potential heterozygosity of the inversions in *C. gardetta*, *C. darwiniana* and *C. cephalidarwiniana* genomes, we performed, with each of their hifiasm haplotypes, another round of genome alignment (minimap2, same parameters), chromosome-level scaffolding (RagTag (Alonge et al., 2022), default parameters), scaffolded genome alignment (minimap2, same parameters) and variant characterization (SyRI, default parameters) against the *C. arcania* reference.

4.4 Selecting large inversions and merging the callsets

We extracted the inversions with a size of at least 100 kb with BCFtools (Danecek et al., 2021) v1.9 for all callsets. We then merged the filtered callsets with the intersect function of BEDtools (Quinlan and Hall, 2010) v2.27.1 using the parameters '-f 0.75 -r', so that two inversions from different callsets were merged if they reciprocally overlapped by a fraction of at least 75% of their length. After removing inversion calls specific to *C. glycerion*, we obtained a total of 20 large inversion calls within the four species-complex.

4.5 Filtering out erroneous calls

In order to filter out false positives and further describe the large inversions obtained in the merged callset, we performed three complementary analyses: (1) detection of the telomeric motif "TTAGG" in inversions breakpoint regions as a signal of assembly scaffolding error, (2) visual validation using alignment dotplots, and (3) breakpoint detection and refinement using orthologous genes.

Sequence from regions spanning 10 kb on each side of the inversion breakpoints were extracted with SAMtools (Danecek et al., 2021) v1.15, and searched for tandem repeats with TandemRepeatFinder (Benson, 1999) v4.09.1 and presence of 200 bp streches of N's indicative of assembly scaffolding point. For all pairs of genomes, dotplots were produced from the minimap2 alignments using the pafR package (`https://github.com/dwinter/pafr`) on each chromosome harbouring an inversion of the merged set. Breakpoints between colinear synteny blocks of orthologous genes identified with Orthofinder v2.5.5 (Emms and Kelly, 2019) were detected and refined with Cassis (Baudet et al., 2010).

We invalidated in total 8 inversions from the merged callset. One of them was identified as a scaffolding error on chromosome 20 upon finding a 2k tandem repetition of the telomeric motif "TTAGG" directly followed by a stretch of 200 N's near one of the inversion's breakpoints. Four other inversion calls were considered as spurious calls from both absence of inversion signal on the dotplots and absence of inversion-like breakpoints from synteny block analysis. Two other inversions, on chromosomes 23 and 24, appeared as more complex rearrangement events (large inverted duplications) on the alignment dotplots with inaccurate breakpoint positions calls. In total

seven invalidated inversions were removed from the final merged set of large inversions. Finally, we merged manually two inversions on chromosome 28 as they had almost identical alignment patterns and position on the dotplots between the species, but were not merged automatically due to an offset of breakpoint position in one callset around a noisy alignment region.

4.6 Estimation of population genomic statistics from SNPs

Genomic information was retrieved for 19 individuals of *C. arcania*, 17 individuals of *C. gardetta*, 16 individuals of *C. cephalidarwiniana* and 9 individuals of *C. darwiniana* from a previously published study on this complex of species (Capblancq *et al.* in prep). Multiple parameters were used to identify patterns of genetic structuring, differentiation and diversity associated with the detected large inversions. The parameter estimates inferred on 50kbp abutting windows were all recovered from Capblancq *et al.* (in prep), which explains in detail how they were calculated. These population level parameters included within population diversity estimates with nucleotide diversity (π) and Tajima's D for each four species, FST between the two parental taxa *C. arcania* and *C. gardetta* and linkage disequilibrium within these two species, as the average r^2 between loci within each window. We also looked at local admixture genetic proportions on each window using K=2 to visualize the proportions of genetic material inherited from *C. arcania* and *C. gardetta* in the two hybrid lineages *C. cephalidarwiniana* and *C. darwiniana*. Finally, we conducted local PCA on each 50kbp window to check for uncommon patterns of genetic distances among samples within and around the identified inversions.

5 Acknowledgements

This work was supported by the French National Research Agency (ANR-20-CE02-0017). We are thankful to the BIPAA bioinformatics platform for hosting the genome and to the GenOuest bioinformatics platform for supporting the calculations.

6 Data Availability

The raw Hifi reads and the genome sequences are available at NCBI under the project PRJXXXXXX. The genome sequences and their annotations are also available publicly at the Bioinformatics Platform for Agroecosystems arthropods (`https://bipaa.genouest.org`).

References

Akbari, E., Naghdi, N., and Motamedi, F. (2006). Functional inactivation of orexin 1 receptors in ca1 region impairs acquisition, consolidation and retrieval in morris water maze task. *Behavioural Brain Research*, 173(1):47–52.

Alonge, M., Lebeigle, L., Kirsche, M., Jenike, K., Ou, S., Aganezov, S., Wang, X., Lippman, Z. B., Schatz, M. C., and Soyk, S. (2022). Automated assembly scaffolding using ragtag elevates a new tomato system for high-throughput genome editing. *Genome biology*, 23(1):258.

Baudet, C., Lemaitre, C., Dias, Z., Gautier, C., Tannier, E., and Sagot, M.-F. (2010). Cassis: detection of genomic rearrangement breakpoints. *Bioinformatics*, 26(15):1897–1898.

Bejarano, F., Luque, C. M., Herranz, H., Sorrosal, G., Rafel, N., Pham, T. T., and Milan, M. (2008). A Gain-of-Function Suppressor Screen for Genes Involved in Dorsal–Ventral Boundary Formation in the Drosophila Wing. *Genetics*, 178(1):307–323.

Benson, G. (1999). Tandem repeats finder: a program to analyze dna sequences. *Nucleic acids research*, 27(2):573–580.

Berdan, E. L., Barton, N. H., Butlin, R., Charlesworth, B., Faria, R., Fragata, I., Gilbert, K. J., Jay, P., Kapun, M., Lotterhos, K. E., Mérot, C., Durmaz Mitchell, E., Pascual, M., Peichel, C. L., Rafajlović, M., Westram, A. M., Schaeffer, S. W., Johannesson, K., and Flatt, T. (2023). How chromosomal inversions reorient the evolutionary process. *Journal of Evolutionary Biology*, 36(12):1761–1782.

Buchfink, B., Reuter, K., and Drost, H.-G. (2021). Sensitive protein alignments at tree-of-life scale using diamond. *Nature Methods*, 18(4):366–368.

Capblancq, T., Després, L., Rioux, D., and Mavárez, J. (2015). Hybridization promotes speciation in coenonympha butterflies. *Molecular Ecology*, 24(24):6209–6222.

Capblancq, T., Mavárez, J., Rioux, D., and Després, L. (2019). Speciation with gene flow: evidence from a complex of alpine butterflies (coenonympha, satyridae). *Ecology and evolution*, 9(11):6444–6457.

Cheng, H., Jarvis, E. D., Fedrigo, O., Koepfli, K.-P., Urban, L., Gemmell, N. J., and Li, H. (2022). Haplotype-resolved assembly of diploid genomes without parental data. *Nature Biotechnology*, 40(9):1332–1335.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., et al. (2021). Twelve years of samtools and bcftools. *Gigascience*, 10(2):giab008.

Emms, D. M. and Kelly, S. (2019). Orthofinder: phylogenetic orthology inference for comparative genomics. *Genome Biology*, 20(1).

Feder, J. L., Egan, S. P., and Nosil, P. (2012). The genomics of speciation-with-gene-flow. *Trends in Genetics*, 28(7):342–350.

Finley, K. D., Taylor, B. J., Milstein, M., and McKeown, M. (1997). dissatisfaction, a gene involved in sex-specific behavior and neural development of drosophila melanogaster. *Proceedings of the National Academy of Sciences*, 94(3):913–918.

Gabur, I., Chawla, H. S., Snowdon, R. J., and Parkin, I. A. P. (2019). Connecting genome structural variation with complex traits in crop plants. *Theoretical and Applied Genetics*, 132(3):733–750.

Goel, M., Sun, H., Jiao, W.-B., and Schneeberger, K. (2019). Syri: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome biology*, 20:1–13.

Gotz, S., Garcia-Gomez, J. M., Terol, J., Williams, T. D., Nagaraj, S. H., Nueda, M. J., Robles, M., Talon, M., Dopazo, J., and Conesa, A. (2008). High-throughput functional annotation and data mining with the blast2go suite. *Nucleic Acids Research*, 36(10):3420–3435.

Guan, D., McCarthy, S. A., Wood, J., Howe, K., Wang, Y., and Durbin, R. (2020). Identifying and removing haplotypic duplication in primary genome assemblies. *Bioinformatics*, 36(9):2896–2898.

Haynes, A. C., Jackson, B., Chapman, H., Tadayyon, M., Johns, A., Porter, R. A., and Arch, J. R. (2000). A selective orexin-1 receptor antagonist reduces food consumption in male and female rats. *Regulatory Peptides*, 96(1):45–51. 10th European Neuropeptide Club.

Heller, D. and Vingron, M. (2020). Svim-asm: structural variant detection from haploid and diploid genome assemblies. *Bioinformatics*, 36(22–23):5519–5521.

Holst, F., Bolger, A., Günther, C., Maß, J., Triesch, S., Kindel, F., Kiel, N., Saadat, N., Ebenhöh, O., Usadel, B., Schwacke, R., Bolger, M., Weber, A. P., and Denton, A. K. (2023). Helixer–de novo prediction of primary eukaryotic gene models combining deep learning and a hidden markov model [preprint]. *BioRxiv*.

Hu, S., Fambrough, D., Atashi, J. R., Goodman, C. S., and Crews, S. T. (1995). The Drosophila abrupt gene encodes a BTB-zinc finger regulatory protein that controls the specificity of neuromuscular connections. *Genes & Development*, 9(23):2936–2948. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

Huang, K. and Rieseberg, L. H. (2020). Frequency, origins, and evolutionary role of chromosomal inversions in plants. *Frontiers in Plant Science*, 11.

Huerta-Cepas, J., Szklarczyk, D., Heller, D., Hernández-Plaza, A., Forslund, S. K., Cook, H., Mende, D. R., Letunic, I., Rattei, T., Jensen, L. J., von Mering, C., and Bork, P. (2018). eggnog 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research*, 47(D1):D309–D314.

Jay, P., Whibley, A., Frézal, L., Rodriguez de Cara, M. A., Nowell, R. W., Mallet, J., Dasmahapatra, K. K., and Joron, M. (2018). Supergene evolution triggered by the introgression of a chromosomal inversion. *Current Biology*, 28(11):1839–1845.e3.

Joron, M., Frezal, L., Jones, R. T., Chamberlain, N. L., Lee, S. F., Haag, C. R., Whibley, A., Becuwe, M., Baxter, S. W., Ferguson, L., Wilkinson, P. A., Salazar, C., Davidson, C., Clark, R., Quail, M. A., Beasley, H., Glithero, R., Lloyd, C., Sims, S., Jones, M. C., Rogers, J., Jiggins, C. D., and ffrench Constant, R. H. (2011). Chromosomal rearrangements maintain a polymorphic supergene controlling butterfly mimicry. *Nature*, 477(7363):203–206.

Kirkpatrick, M. (2010). How and why chromosome inversions evolve. *PLoS Biology*, 8(9):e1000501.

Kulkarni, N. H., Yamamoto, A. H., Robinson, K. O., Mackay, T. F. C., and Anholt, R. R. H. (2002). The DSC1 Channel, Encoded by the smi60E Locus, Contributes to Odor-Guided Behavior in Drosophila melanogaster. *Genetics*, 161(4):1507–1516.

Legeai, F., Romain, S., Capblancq, T., Doniol-Valcroze, P., Joron, M., Lemaitre, C., and Després, L. (2024). Chromosome-Level Assembly and Annotation of the Pearly Heath Coenonympha arcania Butterfly Genome. *Genome Biology and Evolution*, 16(3):evae055.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100.

Li, H. and Durbin, R. (2024). Genome assembly in the telomere-to-telomere era. *Nature Reviews Genetics*, 25(9):658–670.

Li, H. and Ralph, P. (2018). Local pca shows how the effect of population structure differs along the genome. *Genetics*, 211(1):289–304.

Li, N., He, Q., Wang, J., Wang, B., Zhao, J., Huang, S., Yang, T., Tang, Y., Yang, S., Aisimutuola, P., et al. (2023). Super-pangenome analyses highlight genomic diversity and structural variation across wild and cultivated tomato species. *Nature Genetics*, 55(5):852–860.

Liu, Y. H., Luo, C., Golding, S. G., Ioffe, J. B., and Zhou, X. M. (2024). Tradeoffs in alignment and assembly-based methods for structural variant detection with long-read sequencing data. *Nature Communications*, 15(1).

Matsuda, S., Harmansa, S., and Affolter, M. (2016). Bmp morphogen gradients in flies. *Cytokine & Growth Factor Reviews*, 27:119–127. Special Issue:Bone/Body Morphogenetic Proteins.

Nattestad, M. and Schatz, M. C. (2016). Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics*, 32(19):3021–3023.

Nosil, P. (2008). Speciation with gene flow could be common. *Molecular Ecology*, 17(9):2103–2106.

O'Donnell, S. and Fischer, G. (2020). Mum&co: accurate detection of all sv types through whole-genome alignment. *Bioinformatics*, 36(10):3242–3243.

Pazhenkova, E. A. and Lukhtanov, V. A. (2023). Whole-genome analysis reveals the dynamic evolution of holocentric chromosomes in satyrine butterflies. *Genes*, 14(2):437.

Quinlan, A. R. and Hall, I. M. (2010). Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842.

Rodgers, R. J., Halford, J. C. G., Nunes de Souza, R. L., Canto de Souza, A. L., Piper, D. C., Arch, J. R. S., Upton, N., Porter, R. A., Johns, A., and Blundell, J. E. (2001). Sb-334867, a selective orexin-1 receptor antagonist, enhances behavioural satiety and blocks the hyperphagic effect of orexin-a in rats. *European Journal of Neuroscience*, 13(7):1444–1452.

Sanders, A. D., Hills, M., Porubský, D., Guryev, V., Falconer, E., and Lansdorp, P. M. (2016). Characterizing polymorphic inversions in human genomes by single-cell sequencing. *Genome Research*, 26(11):1575–1587.

Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., and Zdobnov, E. M. (2015). Busco: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19):3210–3212.

Smadja, C. M. and Butlin, R. K. (2011). A framework for comparing processes of speciation in the presence of gene flow. *Molecular Ecology*, 20(24):5123–5140.

Strudwick, X. L. and Cowin, A. J. (2020). Multifunctional roles of the actin-binding protein flightless i in inflammation, cancer and wound healing. *Frontiers in Cell and Developmental Biology*, 8.

Thompson, M. J. and Jiggins, C. D. (2014). Supergenes and their role in evolution. *Heredity*, 113(1):1–8.

Weischenfeldt, J., Symmons, O., Spitz, F., and Korbel, J. O. (2013). Phenotypic impact of genomic structural variation: insights from and for human disease. *Nature Reviews Genetics*, 14(2):125–138.

Wellenreuther, M. and Bernatchez, L. (2018). Eco-evolutionary genomics of chromosomal inversions. *Trends in ecology & evolution*, 33(6):427–440.

Wellenreuther, M., Mérot, C., Berdan, E., and Bernatchez, L. (2019). Going beyond snps: The role of structural genomic variants in adaptive evolution and species diversification. *Molecular ecology*, 28(6):1203–1209.

Yan, D. and Perrimon, N. (2015). spenito is required for sex determination in drosophila melanogaster. *Proceedings of the National Academy of Sciences*, 112(37):11606–11611.

Yeaman, S. (2013). Genomic rearrangements and the evolution of clusters of locally adaptive loci. *Proceedings of the National Academy of Sciences*, 110(19).

Zhou, Y., Yu, Z., Chebotarov, D., Chougule, K., Lu, Z., Rivera, L. F., Kathiresan, N., Al-Bader, N., Mohammed, N., Alsantely, A., et al. (2023). Pan-genome inversion index reveals evolutionary insights into the subpopulation structure of asian rice. *Nature Communications*, 14(1):1567.

V Inversions In Pangenome Graphs

In this chapter

1 Introduction 67
Paper: Investigating the topological motifs of inversions in pangenome graphs 68

1 Introduction

After identifying the large *Coenonympha* inversions with a classical approach, I tried to determine whether these same inversions could be identified in pangenome graphs. A pangenome graph presents at least two advantages over reference-based analyses on the *Coenonympha* study case. Firstly, it allows to consider both parental species genomes on the same level of importance rather than having to arbitrarily choose one over the other as reference-genome. Secondly, it allows to directly analyze the inversions between the four genomes simultaneously, rather than having to compare all six pairs of genomes, which involves switching the reference genome used and complexifies the merging of the inversion call sets. I noticed that discovering inversions from pangenome graph's topology was not immediate with the current state of the art tools available, and so I started to work on a method to do so. Additionally, as pangenome graphs are initially designed to represent intra-species genomic variation, it was unclear as to how well state of the art pipelines would handle the level of genomic divergence between the four *Coenonympha* species. Indeed, I detected very few of the Coenonympha inversions in the pangenome graphs, which led me to design several simulated datasets in order to identify the real causes of the lack of inversion detection.

Under the hypothesis that inversions can either be represented explicitly in a bubble's paths or need re-alignment to be identified, I developed a tool to annotate both topology types among pangenome graph bubbles. By testing my tool on pangenome graphs constructed from the four state of the art pipelines, both with the simulated datasets and real data (human and *Coenonympha*), I demonstrated

that the pipelines strongly differ in their way to represent inversions, as well as their capacity to precisely represent them.

This chapter is written in the form of a paper, which is in its way to be submitted to a journal in bioinformatics in the upcoming months. I contributed to the majority of the results presented in this paper, which are the methodological development and implementation of the annotation tool, the production of the simulated datasets, the construction of the simulated and *Coenonympha* pangenome graphs, as well as the analysis of the inversion annotation results on all graphs. I presented an early version of this contribution at the *International Environmental and Agronomic Genomics Symposium* of 2024 in Toulouse (France).

Investigating the topological motifs of inversions in pangenome graphs

Sandra Romain[1], Siegfried Dubois[1], Fabrice Legeai[1,2], and Claire Lemaitre[1]

[1] *Inria, CNRS, IRISA, University of Rennes, 35000 Rennes, France*
[2] *IGEPP, INRAE, Institut Agro, University of Rennes, 35653 Le Rheu, France*

1 Introduction

The recent increase in the number of high-quality assembled genomes for a same species call for a switch in how we represent the genomic reference in genetic diversity analyses, from a single linear reference genome to variation and pangenome graphs. The interest of pangenome graphs resides in their representation of genomic variation between multiple genomes, allowing a diminution of reference bias when mapping reads, thus improving variant discovery and genotyping for the whole range of variant sizes and types (Garrison et al., 2018; Hickey et al., 2024). The current tools for pangenome graph construction rely on genome alignments, either by iterative pairwise alignments from the reference genome for Minigraph (Li et al., 2020) and its derived pipeline Minigraph-Cactus (Hickey et al., 2024), by reference-free pairwise alignments for PGGB (Garrison et al., 2023), or by tree-guided alignments for Progressive Cactus (Armstrong et al., 2020). The standard pipeline in pangenome-graph based variant studies, performed in recent studies such as on bovine (Leonard et al., 2023), wild grape (Cochetel et al., 2023) and human (Liao et al., 2023) genomes, comprises the construction of a graph from a collection of genomes, the identification of variants in the graph and their genotyping in other individuals or populations from read mapping onto the graph. The types of represented variants in the graphs differ between the tools, as Minigraph only aims to represent Structural Variants (SVs), typically variants that are larger than 50 bp, while the other tools also represent smaller variants in their graphs. Variant identification in pangenome graphs consists in analysing the graph topology to find topological motifs called *bubbles*. Bubbles are formed when (at least) two genome paths diverge in the graph due to sequence differences, and meet again further on after the differing portion. For instance, a SNP is represented in the graph by a small bubble where each allele carried in the input genomes forms a diverging path through a node of length 1 bp, while a SV is represented by a larger bubble where at least one of the diverging paths has a length of at least 50 bp. Pangenome graphs may also contain more complex bubbles such as nested bubbles, which can be the result of nested variants (*e.g.* small variants inside an insertion). The main tools for bubble detection are *vg deconstruct* (Liao et al., 2023) (for Minigraph-Cactus, PGGB and Progressive Cactus graphs) and *gfatools* (Li, 2019) (for Minigraph graphs). Although these tools report essential information on the variant bubbles detected in the graphs (*e.g.* position, alleles), they do not annotate the type of the variants detected. Without annotation of the bubbles' variant category, it is difficult to count the number, type and size distribution of the SVs represented in a pangenome graph. The annotation of SNP bubbles does not pose much of a problem as they

have distinctive and exclusive allele path sizes of 1 bp. Likewise, bubbles having one of their allele paths with a size of 0 bp, a distinctive feature of insertions and deletions, can also be confidently annotated. However, all the remaining bubbles, in particular those with large sized paths, are more challenging to confidently annotate as a specific SV type, as their path size may correspond to more than one SV type. The large size of SVs makes them also more likely to contain nested bubbles, which is not that straightforward to represent in a VCF format. Vcfwave, from the vcflib suite (Garrison et al., 2022), can re-align the allele sequences of the large bubbles reported in the *vg deconstruct* VCF to break down all their nested bubbles into a series of insertions, deletions and SNPs.

Among Structural Variants, inversions happen to be less frequent in genomes than other types of variants such as insertions and deletions, but can reach particularly large sizes. In the human genome, inversions account for 0.4% of the diploid genome, with a median size of 17.6 kbp and a maximum size over 5 Mbp (Porubsky et al., 2022). Inversions, among genomic variants, have an exceptional potential to promote adaptation and speciation. When in heterozygous state, they can locally reduce the effective rate of recombination in genomes, which can lead to the spread of adaptative allele combinations in populations or to genetic isolation through mutation accumulation (Kirkpatrick, 2010; Huang and Rieseberg, 2020; Berdan et al., 2023). Large inversions in particular, potentially including more genes, may have a higher probability of capturing non neutral allele combinations (Wellenreuther and Bernatchez, 2018). However, they are known to be more challenging to characterize and genotype than other types of SVs (through sequence based approaches) due to complex and often repetitive genomic context (Liu et al., 2024; Sanders et al., 2016). While a general improvement of structural variant analysis using pangenome graphs has been reported, inversions are often overlooked in pangenome graphs benchmarks and analyses. In Liao et al. (2023) and Leonard et al. (2023) benchmarks of human and bovine pangenomes or Cochetel et al. (2023) analysis of the wild-grape pangenome, SVs were either evaluated as a whole or only insertions and deletions were reported. Only the Hickey et al. (2024) benchmark paper reports some inversion counts, but only for the *Drosophila* pangenome. This could very well be a reflection of the difficulty of distinguishing the different SV types among graph bubbles of large size.

Because accurate representation of all types of variants is a crucial component to make the most out of pangenome graphs, there is a need to assess how inversions are handled by current state of the art pangenome graph pipelines (construction and analysis tools).

In this paper, we explore how inversions are represented in pangenome graphs, as well as the factors impacting their representation and thus the ability to detect them in these graphs. To this end, we constructed pangenome graphs on several simulated genome sets, as well as on real human and butterfly sets of genomes, using the state of the art pipelines. We propose different topologies for inversions and quantified them starting from the bubbles of the graphs detected by the commonly used *vg deconstruct* tool. We found that each pangenome graph pipeline presents a unique pattern of inversion topologies distribution, and that inversions are not always represented explicitly in the paths of the bubbles and need local re-alignment to be identified among the graph bubbles. We propose INVPG-annot, a new tool to annotate among the variant bubbles identified in a pangenome graph those that are likely to represent inversion variants.

2 Material and Methods

2.1 Expected topologies of inversions

In a pangenome graph, the nodes are labeled with genomic sequences, and the directed edges between nodes represent the adjacencies of the sequences in genomes. The nodes can be accessed in two orientations, $forward$ when accessing the node's label sequence or $reverse$ when accessing the reverse complement of the node's label sequence. The paths (succession of nodes and their traversal orientation) in the graph represent the haplotypes used to construct the graph. In such a graph, a shared sequence between several genomes at a given locus is represented as a single node that is traversed by several paths, while a genomic variant is represented by distinct nodes connected in a particular topological motif, called a *bubble*. A bubble is a connected portion of the graph (subgraph) where two nodes, referred as a source node and a sink node, are linked by at least two distinct paths. A single bubble can represent a single variant or several overlapping or nested variants, with all distinct paths between the source and sink nodes identifying all possible alleles or allele combinations of the underlying variant(s).

Contrary to other SV types (*e.g.* insertions, deletions), an inversion does not imply a modification of sequence content between alleles. As such, in the simplest event of inversion (without any inner sequence variation), we can expect both allele sequences to be represented as a single node n_I in the graph, that is traversed in opposite directions (forward and reverse) by the ancestral (p_A) and inverted (p_I) allele paths (Fig. 1a). As inversions can cover large portions of the genome, it is most likely inversion loci harbour inner smaller variants (*e.g.* SNPs, indels). If the graph building algorithm represents small variants in the graph, those will break the inversion locus into multiple nodes and the list of nodes traversed by p_A and p_I will differ (a simple example with one SNP is illustrated in Fig. 1b). Nevertheless, we expect that the balance between the cumulative length of inside variants and the total length of the inversion should allow for the existence of nodes that are common and traversed in opposite directions between p_A and p_I. In both simplistic and realistic cases, the inversion event should be explicitly represented by the paths of the bubble, and so we define this topology type as a "path-explicit" inversion representation.

However, if an inversion is not 'recognized' either during the genome alignment step or during the PG graph construction, both ancestral and inverted alleles could be represented as unrelated sequences in the graph. In the simplest cases with two haplotypes, the locus could be represented with two entirely disjoint paths, one traversing a node n_B carrying the ancestral version of the sequence, the other traversing a node n_Z carrying the inverted version of the sequence (Fig. 1 d,e). In this case, in order to identify the inversion, the sequences of the two paths must be compared, and so we define this topology type as an "alignment-rescued" inversion representation.

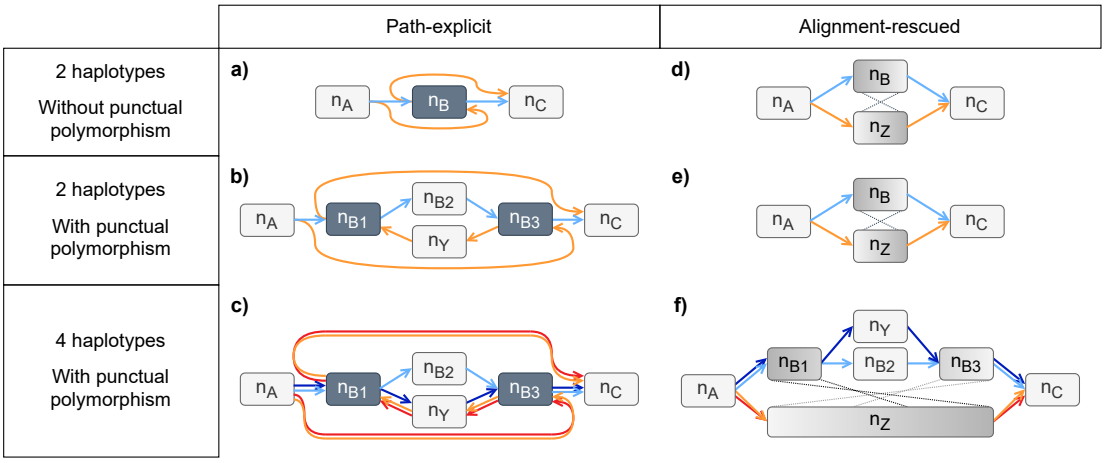

Figure 1: Illustration of the expected types of bubble topologies for inversions in pangenome graphs. Arcs of the graphs are colored according to the presence (orange and red) or absence (blue) of the inversion. The examples on the left illustrate cases where an inversion is explicitly represented in the paths of a bubble ("path-explicit"), with the dark grey colored nodes being the ones that are traversed in both directions **(a, b, c)**. The examples on the right illustrate cases where the two alleles of an inversion are represented by distinct nodes and paths ("alignment-rescued"), with grey-shaded nodes indicating pairs of nodes that are the reverse complement of each other **(d, e, f)**. For each expected topology type, different levels of complexity of the graph are presented: a first level with two haplotypes and no punctual polymorphism inside the inversion **(a, d)**, a second level with two haplotypes and a SNP between the two inversion alleles (**b, e**), and a third level with four haplotypes, one of which (*dark blue*) carries the reference allele of the inversion and the alternative allele of the SNP, meaning that the SNP alleles are not in perfect linkage disequilibrium with the inversion (**c, f**).

2.2 Inversion annotation method

We now present our method to identify among the bubbles found in a pangenome graph, the ones that correspond to these topologies (see an overview in Fig. 2). Our method starts with bubbles that have already been found and aims at annotating them, since the main pangenome graph pipelines (*i.e.* Minigraph-Cactus and PGGB) output bubbles at the same time as they construct the graph. The graph bubbles are reported in the conventional VCF format (in a version adapted to pangenome graphs) by applying the *vg deconstruct* tool on the graph. The pangenome VCF reports the classical information of start position of the variant on a chosen reference genome, and the reference and allele sequences. Additional information specific to the graph is also given by *vg deconstruct*, such as the bubble's source and sink node IDs; the path of each allele of the bubble, given as an ordered and oriented list of nodes, and in the case of nested bubbles, the ID of the parent bubble and the level of the bubble. As those VCFs readily provide an extensive description of the bubble reference and alternative paths, we opted to use them as a base on which to annotate inversions among the bubbles.

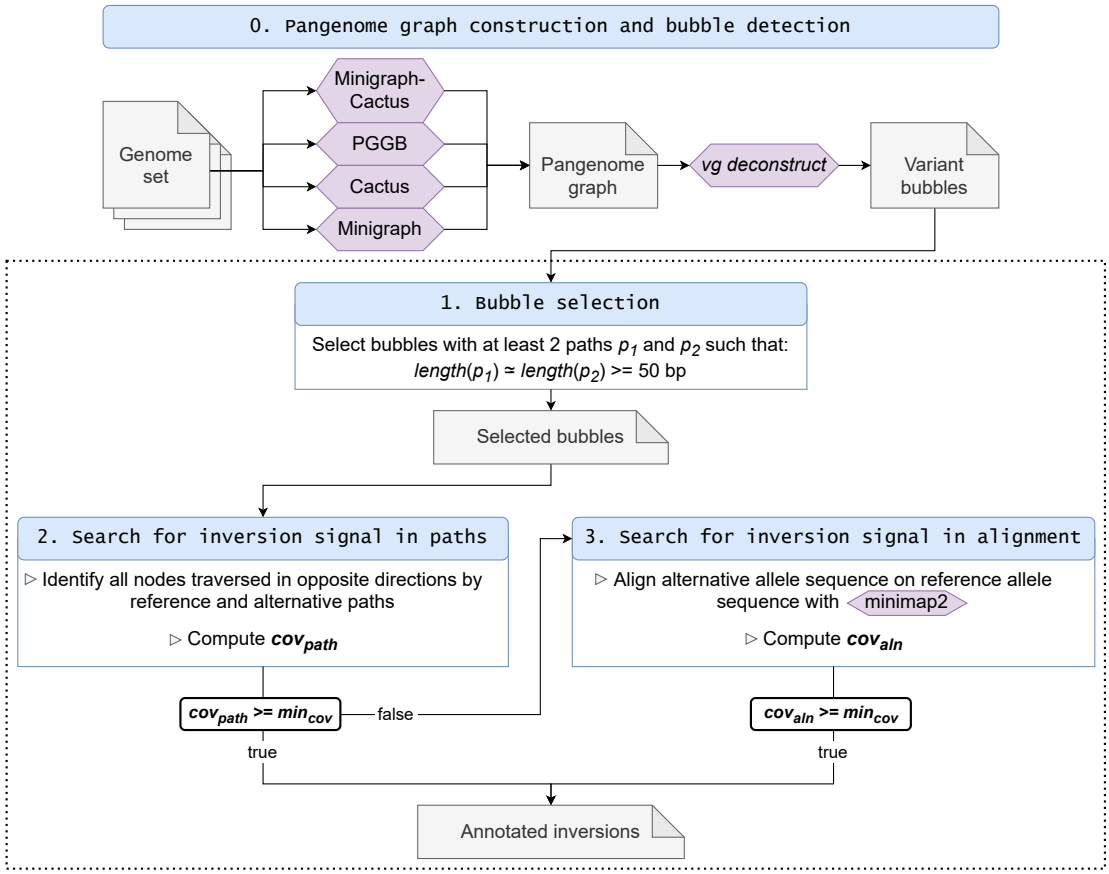

Figure 2: Illustration of the inversion annotation pipeline and method (dotted box). A pangenome graph is constructed from a genome set and analyzed by vg to obtain a list of the variant bubbles represented in the graph. The bubble detection step is included in the pipelines of MGC and PGGB, but must be done additionally when using Progressive Cactus or Minigraph. A first selection of bubbles is done to keep only potential inversion bubbles. These bubbles are then searched for path-based and alignment-based inversion signals.

2.2.1 Bubble selection

In order to limit the amount of bubbles to analyze, we select bubbles resembling to balanced SVs by first extracting bubbles with at least two alleles such that their length are similar and both $\geq$ 50bp. We considered two alleles a_0 and a_1, of length l_0 and l_1, to be of similar length if $|l_0 - l_1| \leq max(l_0, l_1) * \alpha$, with $0 \leq \alpha \ll 1$. The parameter α allows for leniency on the tolerated length difference of inversion alleles, as such difference can originate from innate biological sequence divergence but also from possible computational artefact. We used $\alpha = 0.01$ for the human dataset, $\alpha = 0.1$ for the Coenonympha dataset and, for simulated datasets, we fixed it according to the simulated punctual divergence ($\alpha = (div + 1)/100$).

2.2.2 Search for inversion signal in paths and alignments

In order to annotate inversions among the selected variant bubbles, we first look for inversion signals in the paths of the bubble. For each bubble, all alternative alleles are treated independently and compared to the reference allele.

We consider a path-based inversion signal as a common node between alternative and reference allele paths that is traversed in opposite ways by the two paths. In order to identify these nodes, we convert the reference and alternative paths into two ordered lists $ord_{ref} = [x_1, .., x_n]$ and $ord_{alt} = [y_1, .., y_m]$ of integer node identifiers, which are positive if the node is traversed in forward and negative if the node is traversed in reverse by the path. Then for each node identifier x_i in ord_{ref}, we save x_i as a path-based inversion signal if $-x_i$ is found in ord_{alt}. For each alternative allele, we calculate a path-based signal score cov_{path}, which is the fraction of the reference allele's length covered by path-based inversion signals, with the following formula:

$$cov_{path} = \frac{\sum_{i=1}^{n} length(x_i) \times \mathbf{1}_{\exists j \text{ such that } y_j = -x_i}}{\sum_{i=1}^{n} length(x_i)}$$

If cov_{path} is greater than a user-defined threshold (min_{cov}, with default value of 0.5) for at least one alternative path of the bubble, the bubble is annotated as a path-explicit inversion. If not, we then look for inversion signals in the alignment of the bubble alternative allele sequences to the reference allele. We consider an alignment-based inversion signal as a reverse alignment between the alternative and reference allele sequences. We align all alternative alleles sequences on the reference allele sequence using minimap2 v2.15 with the parameters '-cx asm20 –cs -r2k'. For each alternative allele, we identify all its reverse alignments on the reference allele and calculate an alignment-based signal score cov_{aln} as the length fraction of the reference allele covered by reverse alignments. Considering $A = (a_1, ..a_k)$ the set of reverse alignments, this gives the following formula:

$$cov_{aln} = \frac{\sum_{a_1}^{a_k} length(a_i)}{\sum_{i=1}^{n} length(x_i)}$$

If cov_{aln} is greater than the min_{cov} threshold for at least one of the alternative paths, the bubble is annotated as an alignment-rescued inversion.

2.2.3 Implementation

The presented method of inversion annotation of pangenome graph bubbles is implemented in Python and is available on github (`https://github.com/SandraLouise/INVPG_annot`). It takes a GFA 1.0 of the pangenome graph as well as the corresponding *vg deconstruct*-like VCF (reporting allele walks through the bubbles) as input, and outputs a BED file containing the set of inversion annotated bubbles along with their topology type in the graph (path-explicit or alignment-rescued).

2.3 Genomic material

2.3.1 Simulated genome datasets

We generated several simulated genome sets with well-controlled inversion variants in order to investigate how inversions are represented in pangenome graphs of increasing complexity. We used the genomic sequence of the chromosome 6 of a butterfly species (*Coenonympha arcania*) of length

20 Mb as first genome of each set (Legeai et al., 2024). We defined a set of 100 non-overlapping inversions, whose coordinates were randomly positioned on this chromosome, and whose sizes were picked uniformly between 50 bp and 1 Mb, for a total size of 14 Mb. We then simulated three synthetic chromosomes containing all these 100 inversions but with additional single nucleotide mutations uniformly introduced at different rates: 0, 1 and 5 % rate. As a result, we obtained three 2-genome sets each corresponding to a different nucleotidic divergence level and including the real chromosome sequence as reference and a synthetic genome with its 100 inversions.

We also produced a genome set containing 10 haplotypes, as pangenome graphs are more oftenly used to represent larger collections of genomes. This set contains the reference chromosome 6 along with 9 synthetic haplotypes at a 0% single nucleotide mutation rate. One of the syntetic haplotype contains the whole set of 100 synthetic inversions, while the 8 remaining ones each contain a random sample of 50 inversions among the 100.

2.3.2 Human dataset

For application on human genomes, we based our genome set construction on the extensive inversion dataset published by (Porubsky et al., 2022), reporting 292 balanced inversions accross the GRCh38 reference and genotyped in 44 individuals. We used the GRCh38 reference (Cole et al., 2008), as well as the diploid genome assemblies of the four individuals NA19240, HG00733, HG03486 and HG02818 published by the HPRC (Wang et al., 2022), which were the only assemblies available among the 44 genotyped genomes. In order to limit the processing cost, we selected a single chromosome to process. We filtered the inversion dataset to remove unbalanced inversions (*i.e.* duplicated inversions) and low confidence inversion calls (without the 'PASS' tag). We finally selected the chromosome 7, of 159 Mb, which with 18 filtered inversions present in the four selected genomes was the most inversion-rich autosome in this dataset. The inversions of chromosome 7 range from 1.0 kb to 1.7 Mb, with a median size of 41.2 kb.

We ran RagTag v2.1.0 (Alonge et al., 2022) to resolve the assemblies' contigs correspondance to GRCh38 chromosome 7, and extracted the corresponding (unscaffolded) contigs in a separate multi-fasta file for each of the 9 haplotypes.

2.3.3 *Coenonympha* butterfly dataset

In order to explore the inversion representation in pangenome graphs of more divergent genome sets, we chose four species of alpine *Coenonympha* butterfly (*C. arcania*, *C. gardetta*, *C. darwniniana* and *C. cephalidarwiniana*). While these species are highly related and interfertile, the nucleotidic divergence between their genome is estimated around 7% using Mash (Ondov et al., 2016). The four genomes were recently assembled, with an haplotype size of 500 Mb, and 12 large ($\geq$ 100 kb) inversions were identified between them by whole genome alignment (Romain *et al.*, in prep).

Among the 7 chromosomes of *C. arcania* bearing inversions, we selected the chromosomes 6, 10, 14, 21 and 26 for a total of 8 large inversions. We chose to exclude the Z sex chromosome and one autosome on which the inversion shows very high repetitive and complex genomic context to limit the complexity of the graphs. We extracted the 5 selected chromosomes in a separate multi-fasta file for each of the 8 haplotypes (*i.e.* two haplotypes per genome) and for the *C. arcania* reference haplotype of the species complex, for a total haplotype size of 83 Mb.

2.4 Construction of the pangenome graphs

For each set of genomes, we constructed four versions of pangenome graphs using Minigraph (Li et al., 2020), Minigraph-Cactus (Hickey et al., 2024), PGGB (Garrison et al., 2023) and Progressive Cactus (Armstrong et al., 2020) to explore the diversity of inversion representation across the state of the art tools.

We ran Minigraph-Cactus v2.8.2 with the parameters '–clip 0 –filter 0' to retain all haplotype paths in the graphs. PGGB v0.6.0, Minigraph v0.21 and Progressive Cactus v2.8.2 were run with default parameters. As it is recommended to use binarized input trees with the more recent versions of Progressive Cactus, we wrote arbitrary trees with uniform branch length to ensure they have the right structure, keeping haplotypes of single individuals closest together in the trees. We recon that the trees should have minimal impact on human and simulated data experiments as genomes are close, and we built the *Coenonympha* genome tree corresponding to Capblancq et al. (2019). We then converted the resulting HAL file of Progressive Cactus to a VG graph format using the *hal2vg* tool (included in Cactus v2.8.1) with the '–noAncestors' parameter. *Vg convert* (Garrison et al., 2018) v1.50.1, was then used with the '-f -W' parameters to obtain the Minigraph and Progressive Cactus graphs in GFA 1.0 format.

For all constructed graphs, we calculated the graphs statistics using pancat (Dubois, 2023) v0.3.0.

Variant bubbles were obtained using *vg deconstruct* for all graphs. The reference genome, used as coordinate system for the output VCF file, was set as GRCh38 for the human dataset, *C. arcania* for the *Coenonympha* dataset, and the reference sequence for the simulated dataset. For PGGB and Minigraph-Cactus, this step was included in their pipeline. For Minigraph and Cactus, we ran *vg deconstruct* with the parameter '-a' to obtain an exhaustive set of bubbles (without any filter).

2.5 Evaluation of inversion detection in the pangenome graphs

We assessed the completeness of the sets of annotated inversion bubbles by calculating a recall value as the percentage of known inversions that were annotated in the graph by at least one of its bubbles. We established the correspondence of the annotated inversions to the true inversions in the datasets based on their position overlap on the reference using bedtools (Quinlan, 2014) v2.27.1. We considered a given inversion as correctly annotated if a reciprocal overlap of at least 50% was found between the inversion and one of the annotated bubbles (*i.e.* the position of the annotated bubble on the reference covers $\geq 50\%$ of the size of the inversion, and vice-versa), using the options '-f 0.5 -r' of the function *bedtools intersect*. We also calculated the number of false positive annotations, which we define as annotated bubbles for which no overlap was found with any of the true inversions (without minimum overlap fraction).

3 Results

3.1 Investigating inversion bubble topologies in most simple simulated graphs

We first investigated how inversions are represented in pangenome graphs that we can imagine to be the easiest instances for inversion detection, that are graphs built from only two genomes differing only by non overlapping and randomly positioned inversions. To do so, we built graphs for the

set of two 20 Mb chromosomes differing by 100 inversions and without any other polymorphism (see Section 2.3.1), using four pangenome graph pipelines Minigraph, Minigraph-Cactus (MGC), Progressive Cactus (Cactus) and PGGB.

3.1.1 Pangenome graph complexity

For such a simple genome set, we expect its pangenome graph to contain few nodes and edges and a total sequence content of exactly 20 Mb, since inversions do not produce any novel sequences, only novel adjacencies. More precisely, the theoretical number of nodes is expected to be $1 + 2N$ and the theoretical number of edges $4N$, for N the number of inversions. In practice, we obtained rather different graph statistics for all four tools (Table 1). Except for Minigraph, all three other tools have more than ten times more nodes and edges than expected, with the largest value for Minigraph-Cactus with more than 200 thousands nodes. Furthermore only Cactus and PGGB have a size close to the expected 20 Mb. Minigraph and MGC graphs contain more than 5 Mb of additional sequences, being either duplicated or more probably present in both forward and reverse-complement versions in the graph.

Table 1: Basic statistics of the pangenome graphs obtained for the simulated genome set with 100 inversions and no punctual polymorphism, for each pangenome graph constructing tool used. The expected size of this graph is 20 Mb with 100 bubbles. The number of bubbles indicates the total number of bubbles reported by vg deconstruct. Large bubbles are those with at least two paths of similar length greater than 50 bp. Inversion bubbles are those annotated as inversion by our method. The Recall column indicates the percentage of simulated inversions (among the 100) that have a significant reciprocal overlap with at least one inversion annotated bubble. ("MGC": Minigraph-Cactus).

PG tool	Graph size (Mb)	# Nodes	# Edges	# Bubbles	# Large bubbles	# Inversion bubbles	Recall (%)
Cactus	19.99	7,275	9,506	3,128	66	64	63
Minigraph	25.49	131	166	48	22	16	5
MGC	25.28	200,869	216,069	15,167	32	28	8
PGGB	20.30	3,145	4,332	5,114	92	88	86

3.1.2 Many inversions are not represented as bubbles in the graphs

We applied *vg deconstruct* to find bubbles in these graphs and then we applied our method to look for inversion signals in these bubbles. We expected 100 bubbles in total in the graphs, as the two haplotypes used to construct the graphs differ by only 100 inversions. Minigraph graph contains half the number of expected bubbles, while Cactus, PGGB and MGC graphs contain respectively 30, 50 and 150 times more bubbles than expected (Table 1). For all the graphs, less than 100 bubbles pass our method's filter on bubble size (*i.e.* at least two paths $\geq$ 50 bp and of similar size), which represent 0.2, 1.8, 2.1 and 46 % of the total number of bubbles in the graphs of Minigraph-Cactus, PGGB, Cactus and Minigraphs, respectively. Most of the large and balanced selected bubbles are annotated as inversions, though in fewer number than expected: respectively 16, 28, 64 and 88 for Minigraph, Minigraph-Cactus, Cactus and PGGB graphs.

We compared the annotated inversions to the real inversions present in our dataset, and obtained relatively low to very low recall values of 86, 63, 8 and 5 % with the PGGB, Progressive Cactus, Minigraph-Cactus and Minigraph pipelines, respectively (Fig. 3A). On the other hand, there was no false-positive annotation, all annotated inversions are positioned at real inversion sites on the reference. However, we observed redundancy in the annotated bubbles with the Minigraph-Cactus graph, in the sense that among the 14 annotated bubbles of the graph, 3 and 5 bubbles corresponded to a same simulated inversion. The size of the inversions did not seem to impact their annotation rate in Minigraph's graph. On the other hand, for all other three graphs, the unannotated inversions generally appear to be among the smaller ones (under 100 kb with Minigraph-Cactus, under 50 kb with Cactus and PGGB) (Fig. 3C).

3.1.3 Several inversion bubble topologies between graph pipelines

Inversion annotation were divided in two types : (i) "path-explicit": the signals of inversion can be directly identified in the graph (ii) "alignment rescued": the inversion has been detected after the alignment of the bubble path sequences. As a result, we observed that the graphs differ in the topology type (Fig. 3B). All inversions detected in the Cactus graph are *path-explicit*, while there are both *path-explicit* and *alignment-rescued* topologies in the other graphs. Interestingly, we found that the inversion topology in PGGB is strongly linked to the inversion size, as all path-explicit inversions are larger than 50 kb, and all alignment-rescued inversion are smaller than this threshold (Fig. 3C). In the case of Minigraph-Cactus, the path-explicit inversions are also larger in size than the alignment-rescued ones (threshold around 500 kb). It seems to be the opposite for Minigraph, although the threshold is not that strict as the inversion size range between the two topology types overlaps around 100 kb.

3.1.4 Accuracy of inversion breakpoints

We measured the accuracy of the inversion breakpoints represented in the pangenome graphs by calculating the offset of both start and end positions of the true positive inversion annotated bubbles compared to the true positions of the inversion in the simulated datasets (Fig 4). An offset of 0 means that the breakpoints of the inversion-annotated bubble correspond exactly to that of the inversion position in the genome, while a greater offset means that the inversion-annotated bubble extends before (start position offset, Fig 4A) or after (end position offset Fig 4B) the inversion true position in the genome.

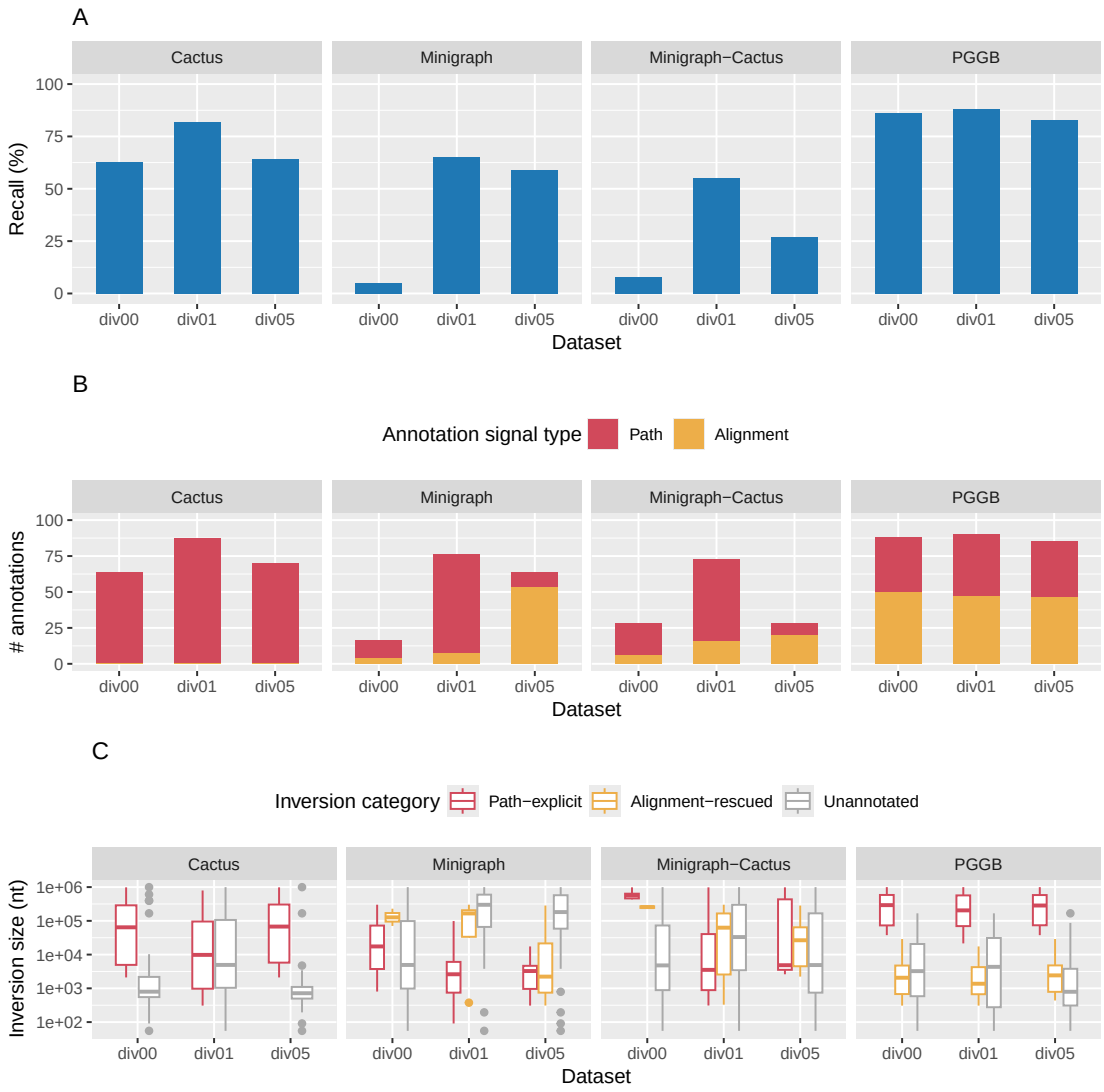

Figure 3: Inversion annotation statistics for the 2-haplotype simulated datasets pangenome graphs depending on the PG constructing tool. (A) Recall of inversion annotation as calculated by the percentage of inversions overlapped by at least one bubble annotated as inversion. A minimum threshold of 50% reciprocal overlap was used. (B) Number of inversion annotations based on a path signal ('path explicit', *red*) or an alignment signal ('alignment-rescued', *orange*). (C) Size distribution of the true inversions annotated by path signal (*red*), annotated by alignment signal (*orange*), or unannotated (*grey*). The datasets shown are 0% SNP ('div00'), 1% SNP ('div01') and 5% SNP ('div05').

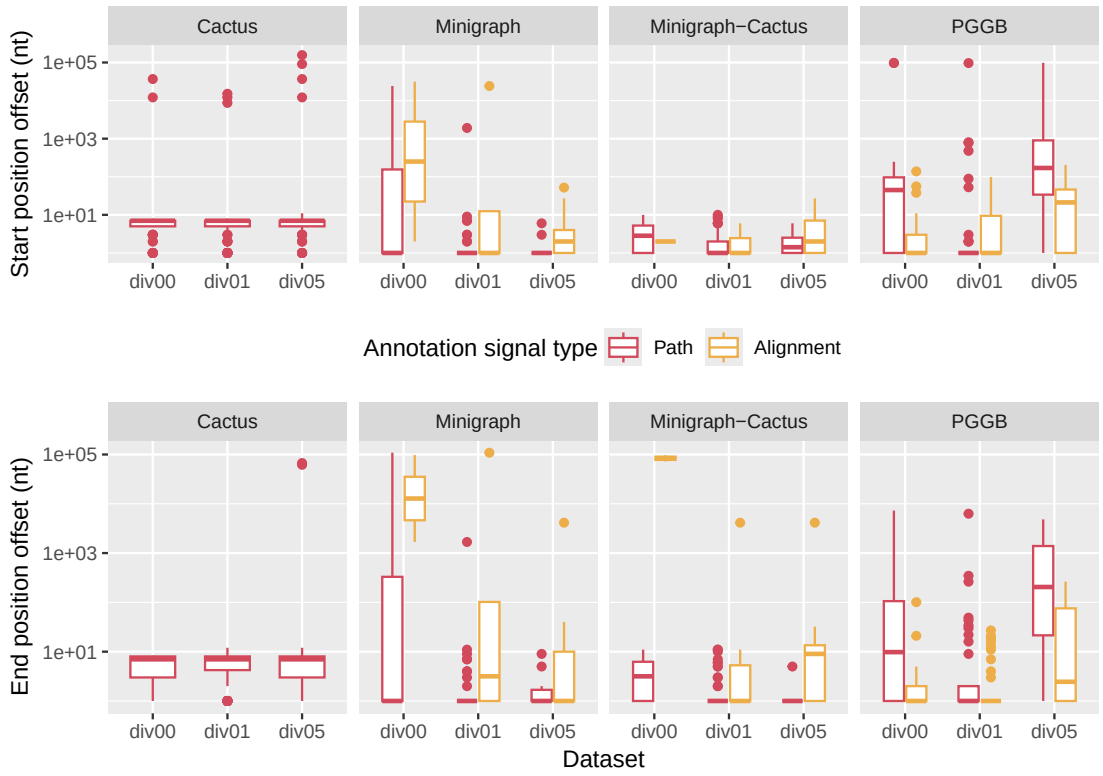

Figure 4: Inversion bubble start and end position offsets in the pangenome graphs. The start position offset (upper panel) represents the offset between the true inversion start and the start position of the inversion-annotated bubble in the graph, and the end position offset (lower panel) represents the offset between the true inversion end and the end position of the inversion-annotated bubble in the graph. The Y axis is log-scaled.

Among all pangenome graphs on simulated datasets, the Cactus graph shows the most accurate annotated inversion positions with an offset mostly smaller than 10 bp. As expected, Miningraph appeared as the less precise tool concerning inversion coordinates. However, for almost all tools, we can observe extreme values of offsets as large as dozens of Kbs. Interestingly, we found that in PGGB graphs, the path-explicit inversion bubbles show an overall greater position offset than the alignment-rescued ones, with differences enlarging the inversions, which means that the bubbles "capture" extra neighbouring sequence.

3.2 Factors impacting inversion topologies in simulated graphs of increasing complexity

Next, we simulated more realistic and complex graphs by increasing the nucleotidic divergence of the input haplotypes and by increasing the number of haplotypes in the graphs. On one hand, we built graphs for the sets of two 20 Mb haplotypes differing both by 100 inversions and 1 or 5 %

SNPs to test the impact of the single nucleotide divergence (see Section 2.3.1). On the other hand, we built graphs for the set of ten 20 Mb haplotypes, differing by varying subsamples of 100 initial inversions, and without any other polymorphism (see Section 2.3.1).

3.2.1 Increasing the level of single nucleotide divergence

As expected, the three pipelines aiming at representing the whole range of variants show an great increase in node, edge and bubbles numbers in their graphs compared to the graphs obtained for genome sets without small polymorphisms (Tab. 2). These are in the same order of magnitude as the number of simulated variants (200,000 and 1 million for 1 and 5 % divergence datasets respectively). Although Minigraph is not meant to represent small polymorphism, the presence of SNPs increases a little bit its graph complexity, with hundreds of additional nodes compared to the graph obtained for the genome set without punctual divergence. Surprisingly, Minigraph and Minigraph-Cactus pipelines produced graphs of size closer to the haplotype size when the nucleotidic divergence increases.

Table 2: Basic statistics of the pangenome graphs obtained for the simulated genome sets with 100 inversions and with varying percentages of punctual polymorphism, for each pangenome graph constructing tool used. All of the genome sets have a haplotype size of 20 Mb. The number of bubbles indicates the total number of bubbles reported by vg deconstruct. The Recall column indicates the percentage of simulated inversions (among the 100) that have a significant reciprocal overlap with at least one inversion annotated bubble. ("MGC": Minigraph-Cactus).

Genome set	PG tool	Graph size (Mb)	# Nodes (10^3)	# Edges (10^3)	# Bubbles	Recall (%)
1% SNP	Cactus	20.33	601.2	800.8	198,952	82
	Minigraph	21.43	0.3	0.3	107	65
	MGC	24.02	864.1	1,144.1	279,837	55
	PGGB	20.18	587.6	783.5	195,771	88
5% SNP	Cactus	21.38	2,902.8	3,853.9	954,364	64
	Minigraph	21.39	0.3	0.3	89	59
	MGC	22.32	2,318.1	3,073.5	755,405	27
	PGGB	21.65	2,803.7	3,740.6	945,302	83

Even more surprising is the sharp increase of inversion annotation recall of Minigraph and Minigraph-Cactus graphs at nucleotidic divergence of 1 % compared to nucleotidic divergence of 0 %, rising from 5% to 65% for Minigraph and from 8% to 55% for Minigraph-Cactus (Fig. 3A). PGGB still obtains the highest recall values, among all tools around 85 %, regardless of the punctual nucleotidic divergence. The increase of nucleotidic divergence from 1 to 5 % does not appear to have a clear impact on recall, except for the Minigraph-Cactus graphs for which the recall drops by 28 % points down to 27 %.

Inversion's topology and breakpoint accuracy in the graphs also seem little related to nucleotidic divergence for all pipelines at the exception of Minigraph, where we observe a switch from a majority of path-explicit topology types at 1 % divergence to a majority of alignment-rescued topology types

at 5 % divergence (Fig. 3B), along with a decrease of start and end position offsets down to less than 10 bp at 5 % divergence (Fig 4).

3.2.2 Increasing the number of haplotypes in the graph

When increasing the number of represented haplotypes in the graph from two to ten, we observe an increased recall of inversion annotation for all pipelines, althouth the number of simulated inversion is not changed (Tab. 3). The highest gains in recall are observed for Minigraph-Cactus (+ 12 % points, but still low) and Cactus (+ 12 % points) pipelines. However, not all 100 inversions could be annotated in any of the graphs, and ten inversions were still missing in the best tool, PGGB.

Table 3: Inversion annotation recall obtained for the 2-haplotype and 10-haplotype simulated genome set with no punctual polymorphism, for each pangenome graph constructing tool used ("MGC": Minigraph-Cactus).

PG tool	Recall (%)	
	2 haplotypes	10 haplotypes
Cactus	63	75
Minigraph	5	11
Minigraph-Cactus	8	20
PGGB	86	90

3.3 Inversion bubble annotation in real pangenome graphs

Finally, we investigated how real inversion variants are represented in pangenome graphs constructed with two real genome datasets. The first dataset used is a human genome dataset containing nine haplotypes for the 159 Mb chromosome 7, for which we would expect to find at least 18 inversions described in Porubsky et al. (2022) (see Section 2.3.2). The second dataset used is a butterfly (*Coenonympha*, Satyrinae) genome dataset containing nine haplotypes along 5 chromosomes, with a total haplotype size of 83 Mb, for which we aimed to annotate 8 large inversions (see Section 2.3.3).

As expected, we obtained graphs of a much higher complexity than for simulated genomes, illustrated in Table 4. Notably, the *Coenonympha* graphs have a striking total size of 2.6 to 4 times the single haplotype size, while the size of the human graphs is of the same order of magnitude than that of the single haplotype. We obtained a very low recall of inversion annotation for both datasets, not more than 25% and 20% for the human and *Coenonympha* datasets, respectively (Tab. 4). Furthermore, no inversion was annotated on the human dataset with the Cactus graph, or on the *Coenonympha* dataset with the Cactus or Minigraph graphs. All annotated inversions but one with the human dataset are path-explicit in the graphs.

The running time and memory requirement of INVPG-annot on the pangenome graphs VCFs constructed on the human dataset and with the four state of the art pipelines are shown in Table 5. INVPG-annot ran in less than one minute on the Cactus and Minigraph VCFs, and in 11 minutes on the Minigraph-Cactus VCF, while it took more than two hours to run on the PGGB VCF. We observed that this running time is correlated to the number of allele alignments performed, which is with little surprise the limiting factor of our tool's running speed. We observed the same correlation for the memory consumption of INVPG-annot.

Table 4: Basic statistics of the pangenome graphs obtained for the human and Coenonympha genome sets, for each pangenome graph constructing tool used ("MGC": Minigraph-Cactus). The human genome set has a haplotype size of 159 Mb, the *Coenonympha* genome set has a haplotype size of 83 Mb.

Genome set	PG tool	Graph size (Mb)	# Bubbles (10^3)	Recall (%)
Human, chr7	Cactus	176.83	25.5	0.0
(9 haplotypes)	Minigraph	162.05	3.3	22.2
	MGC	179.11	594.2	22.2
	PGGB	168.33	663.2	16.7
Coenonympha	Cactus	333.80	6,135	0.0
(9 haplotypes)	Minigraph	228.98	63	0.0
	MGC	321.93	4,751	25.0
	PGGB	289.07	4,235	25.0

Table 5: Total CPU time and maximum memory requirement of inversion annotation from the *vg deconstruct* VCF files obtained from the four human pangenome graphs. All annotations were run on 1 CPU thread.

PG tool	Total CPU time (min)	Max memory (Mo)
Cactus	0.6	1.2
Minigraph	0.2	1.2
Minigraph-Cactus	11.0	533.8
PGGB	146.2	1,946.5

4 Discussion

4.1 Very contrasted inversion handling between the pangenome graph constructing tools

The simulated datasets highlighted highly contrasted representations of inversion variants between Cactus, Minigraph-Cactus and PGGB. Firstly, we found very few inversions with the expected bubble topologies with Minigraph-Cactus compared to Cactus and PGGB, with an average of 19% against 64.3% and 81.7%. As the inversion signals required to annotate bubbles as inversions are not very stringent (at least 50 % of the bubble's reference allele comprised in segments with opposite path traversals or in reverse alignments), we hypothetize that some unannotated inversions are not fully contained in input bubbles. In particular in the case of Minigraph-Cactus, we suspect that inversion loci that are firstly represented as unrelated sequences at the Minigraph step could then be split at the Cactus step and end up represented as chains of bubbles in the graph, thus spread over multiple high-level bubbles in the VCF. It could also be that some specific bubble topologies in the graph are not reported by the bubble detection tool, as we additionally observed on other graphs that *gfatools bubble* reported more bubbles than *vg deconstruct* VCF. Unfortunately, we could not find sufficient details on both bubble detection algorithms to confirm this hypothesis. An

possible improvement of the method would be to directly scan the graph for the target topologies in order to bypass the constraints of the bubbles detection tools (*vg deconstruct* or *gfatools bubble*).

The pangenome graph constructing tools also showed different patterns of inversion topology distribution, and the need to re-align bubble alleles proved to be especially relevant for inversions of smaller size (below 50 kb) in the PGGB graphs. We found that the inversion size threshold between path-explicit and alignment-rescued topologies seems to be mainly dependant on the parameter set used for the genome alignment step with wfmash ('-s', '-l' and '-c'). When wfmash outputs large alignments encompassing smaller inversions, these inversions are not detected by seqwish which analyses only CIGAR strings to find sequence variation. After testing different wfmash parameters values, we found that the default values of '-s', '-l' and '-c' were those giving the best inversion annotation recall with the simulated datasets.

4.2 The issue with inversions represented as unrelated sequences

While inversions represented as unrelated sequences in the graph (*i.e.* with distinct nodes for the ancestral and inverted version of the sequence) can still be recovered if there is at least one bubble containing the entire locus, they need additional alignment to be annotated, which can become a computational burden for more large and complex graphs (more than 2 hours for the annotation step only with the PGGB human pangenome graph's bubbles). Moreover, those sequences with a same origin and position are represented twice in the graph, which could raise issues with downstream processing and analyses of the graph. For example, if there are small variants (e.g. SNPs) inside such inversions that one wishes to genotype, the question arises as of how these are represented and detected (on one or both of the inversion distinct paths, or simply missed) and how accurate the mapping of reads would be between those distinct paths. Reads originating from the inverted locus could be mapped to either of the paths, generating multi-mapping problems. As a conclusion, these inversions not explicitely represented in the graph need to be recovered as a path-explicit topology to reduce the redundancy and ambiguity of the pangenome graphs, otherwise limiting their usage.

4.3 A new tool to annotate inversions and inspect their topology in pangenome graphs

Here we presented a method and tool which detects inversions in pangenome graphs through the analysis of variant bubbles such as those output by Minigraph-Cactus and PGGB, or *via vg deconstruct*. Its stringency on inversion allele size variation can be adapted to the diversity of the pangenomes, and the stringency of required coverage for inversion signal can also be set to the desired value, although these parameters can affect both computing time and annotation precision. With a higher inversion size variation and a graph rich in unrelated-sequence-like bubbles, the amount of alignment performed could slow the annotation process. While a lower stringency on required coverage of inversion signal helps recover inversions "hidden" in larger bubbles, it can also decrease the precision of the annotation. As the method looks for inversion signal inside each bubble, its annotation recall is affected by how the tool used to construct the graph represents inversions, but also may depend on the bubble detection algorithm used. With the provided type of topology for each detected inversion, we argue that it could be used to eventually perform targetted modifications of the pangenome graph in order to transform alignment-rescued inversions into path-explicit.

5 Acknowledgements

6 Data Availability

References

Alonge, M., Lebeigle, L., Kirsche, M., Jenike, K., Ou, S., Aganezov, S., Wang, X., Lippman, Z. B., Schatz, M. C., and Soyk, S. (2022). Automated assembly scaffolding using ragtag elevates a new tomato system for high-throughput genome editing. *Genome biology*, 23(1):258.

Armstrong, J., Hickey, G., Diekhans, M., Fiddes, I. T., Novak, A. M., Deran, A., Fang, Q., Xie, D., Feng, S., Stiller, J., et al. (2020). Progressive cactus is a multiple-genome aligner for the thousand-genome era. *Nature*, 587(7833):246–251.

Berdan, E. L., Barton, N. H., Butlin, R., Charlesworth, B., Faria, R., Fragata, I., Gilbert, K. J., Jay, P., Kapun, M., Lotterhos, K. E., et al. (2023). How chromosomal inversions reorient the evolutionary process. *Journal of evolutionary biology*, 36(12):1761–1782.

Capblancq, T., Mavárez, J., Rioux, D., and Després, L. (2019). Speciation with gene flow: evidence from a complex of alpine butterflies (coenonympha, satyridae). *Ecology and evolution*, 9(11):6444–6457.

Cochetel, N., Minio, A., Guarracino, A., Garcia, J. F., Figueroa-Balderas, R., Massonnet, M., Kasuga, T., Londo, J. P., Garrison, E., Gaut, B. S., et al. (2023). A super-pangenome of the north american wild grape species. *Genome Biology*, 24(1):290.

Cole, C. G., McCann, O. T., Collins, J. E., Oliver, K., Willey, D., Gribble, S. M., Yang, F., McLaren, K., Rogers, J., Ning, Z., Beare, D. M., and Dunham, I. (2008). Finishing the finished human chromosome 22 sequence. *Genome Biology*, 9(5):R78.

Dubois, S. (2023). pancat. `https://github.com/Tharos-ux/pancat`.

Garrison, E., Guarracino, A., Heumos, S., Villani, F., Bao, Z., Tattini, L., Hagmann, J., Vorbrugg, S., Marco-Sola, S., Kubica, C., et al. (2023). Building pangenome graphs. *bioRxiv*, pages 2023–04.

Garrison, E., Kronenberg, Z. N., Dawson, E. T., Pedersen, B. S., and Prins, P. (2022). A spectrum of free software tools for processing the vcf variant call format: vcflib, bio-vcf, cyvcf2, hts-nim and slivar. *PLoS Computational Biology*, 18(5):e1009123.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., et al. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36(9):875–879.

Hickey, G., Monlong, J., Ebler, J., Novak, A. M., Eizenga, J. M., Gao, Y., Marschall, T., Li, H., and Paten, B. (2024). Pangenome graph construction from genome alignments with minigraph-cactus. *Nature biotechnology*, 42(4):663–673.

Huang, K. and Rieseberg, L. H. (2020). Frequency, origins, and evolutionary role of chromosomal inversions in plants. *Frontiers in plant science*, 11:296.

Kirkpatrick, M. (2010). How and why chromosome inversions evolve. *PLoS biology*, 8(9):e1000501.

Legeai, F., Romain, S., Capblancq, T., Doniol-Valcroze, P., Joron, M., Lemaitre, C., and Després, L. (2024). Chromosome-Level Assembly and Annotation of the Pearly Heath Coenonympha arcania Butterfly Genome. *Genome Biology and Evolution*, 16(3):evae055.

Leonard, A. S., Crysnanto, D., Mapel, X. M., Bhati, M., and Pausch, H. (2023). Graph construction method impacts variation representation and analyses in a bovine super-pangenome. *Genome Biology*, 24(1):124.

Li, H. (2019). gfatools. `https://github.com/lh3/gfatools`.

Li, H., Feng, X., and Chu, C. (2020). The design and construction of reference pangenome graphs with minigraph. *Genome biology*, 21:1–19.

Liao, W.-W., Asri, M., Ebler, J., Doerr, D., Haukness, M., Hickey, G., Lu, S., Lucas, J. K., Monlong, J., Abel, H. J., et al. (2023). A draft human pangenome reference. *Nature*, 617(7960):312–324.

Liu, Y. H., Luo, C., Golding, S. G., Ioffe, J. B., and Zhou, X. M. (2024). Tradeoffs in alignment and assembly-based methods for structural variant detection with long-read sequencing data. *Nature Communications*, 15(1).

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: fast genome and metagenome distance estimation using minhash. *Genome biology*, 17:1–14.

Porubsky, D., Höps, W., Ashraf, H., Hsieh, P., Rodriguez-Martin, B., Yilmaz, F., Ebler, J., Hallast, P., Maggiolini, F. A. M., Harvey, W. T., et al. (2022). Recurrent inversion polymorphisms in humans associate with genetic instability and genomic disorders. *Cell*, 185(11):1986–2005.

Quinlan, A. R. (2014). Bedtools: the swiss-army tool for genome feature analysis. *Current protocols in bioinformatics*, 47(1):11–12.

Sanders, A. D., Hills, M., Porubský, D., Guryev, V., Falconer, E., and Lansdorp, P. M. (2016). Characterizing polymorphic inversions in human genomes by single-cell sequencing. *Genome Research*, 26(11):1575–1587.

Wang, T., Antonacci-Fulton, L., Howe, K., Lawson, H. A., Lucas, J. K., Phillippy, A. M., Popejoy, A. B., Asri, M., Carson, C., Chaisson, M. J. P., Chang, X., Cook-Deegan, R., Felsenfeld, A. L., Fulton, R. S., Garrison, E. P., Garrison, N. A., Graves-Lindsay, T. A., Ji, H., Kenny, E. E., Koenig, B. A., Li, D., Marschall, T., McMichael, J. F., Novak, A. M., Purushotham, D., Schneider, V. A., Schultz, B. I., Smith, M. W., Sofia, H. J., Weissman, T., Flicek, P., Li, H., Miga, K. H., Paten, B., Jarvis, E. D., Hall, I. M., Eichler, E. E., and Haussler, D. (2022). The human pangenome project: a global resource to map genomic diversity. *Nature*, 604(7906):437–446.

Wellenreuther, M. and Bernatchez, L. (2018). Eco-evolutionary genomics of chromosomal inversions. *Trends in ecology & evolution*, 33(6):427–440.

VI Discussion And Perspectives

In this chapter

Genotyping structural variants using graphs 87
- A fast and accurate long-read SV genotyper on variation graph 87
- Managing imprecise breakpoints. 89
- Adaptation of SVJedi-graph's method to pangenome graphs 90
- Uncompleteness of structural variant genotyping evaluation 90

Inversion discovery: reference genome *versus* pangenome graph 91
- Reference-based SV discovery tools 91
- Inversion annotation from pangenome graphs 92
- Final thoughts on inter-specific inversion discovery 93

Genotyping structural variants using graphs

The problem of SV genotyping often involves counting the number of reads supporting each allele of a given SV. In existing methods using read mapping on the reference genome, this read count can be biased towards the reference allele. Proximity between SVs can also impair the mapping of reads when the genome or alleles are represented as linear sequences, leading to a decrease of genotyping rate. These two limitations can be overcome by using a variation graph to represent the genome with its known variants. This strategy was exclusively applied in short-read based genotypers. As long reads have proven to yield better mapping results in repeated regions, which harbour a large portion of SVs in genomes, I developed a long-read SV genotyping method using this variation-graph representation.

A fast and accurate long-read SV genotyper on variation graph

I produced a genotyping method able to predict more accurate genotypes than all compared long-read genotypers. Based on the difference of genotyping accuracy with SVJedi and the fact that all parts of my method unrelated to the use of variation graph were heavily inspired from SVJedi, I believe the variation graph representation to be the main factor contributing to this improved accuracy. The variation graph representation is interesting in particular for the genotyping of

close and overlapping SVs. My results on simulated data showed that either the accuracy or rate of genotype predictions from linear representation-based tools starts to decline when variants are as close as 1 kbp to one another. This effect of SV closeness was not apparent in these tool's benchmarks, as the unanimously used high quality benchmark dataset (GIAB HG002) only contains SVs selected to be distant of at least 1 kbp. This dataset does not reflect reality, in practice, SV discovery tools often output close and overlapping variants. In this sense, my method could provide a significant improvement for the genotyping of SVs in SV dense regions. However, its genotyping accuracy strongly depends on the precision of SV breakpoints predicted by the discovery tools, and so in turn depends on the ability of SV discovery tools to infer accurate breakpoints in such regions.

Another asset of my method is its fast computing time. For instance, its overall genotyping time was six times faster than all other state of the art tools on human data. This outcome was not expected, as mapping on graphs is more computationally intensive than mapping on linear sequences. In the very first version of SVJedi-graph (1.0.0), the read mapping was performed using GraphAligner, and the genotyping time was indeed higher than reference-based genotypers. By switching to the minigraph mapper and its option to bypass the base-level alignment of the long reads in version 1.1.0, I was able to drastically cut the computing time and render my method's performances competitive against state of the art tools. On the other hand, minigraph could possibly be less sensitive than GraphAligner in instances with small nodes, as it needs at least one minimizer in nodes in order to map reads on them. The sole requirement of the mapping position of the reads in the graph to deduce the supported allele is thus another advantage of the variation graph compared to 'force-calling' methods, which need the complete read alignments on the reference genome to genotype variants.

This method produced the first tool to genotype structural variants with long reads using a variation graph, SVJedi-graph. I worked on rendering this tool easily accessible to users by making it available both on a documented Git repository[1] and as a BioConda package[2]. Both supports have been continuously updated through my thesis with bug fixes and algorithmic improvements. Up to the writing of this manuscript, SVJedi-graph has been downloaded 5295 times in total *via* its BioConda package. Its algorithm inspired Somrit (D'Costa and Simpson, 2023), a toolkit made for the detection of somatic mobile element insertion from long reads. I also got feedback from users on its Git repository, and some of the applications of my tool on biological data that I am aware of are those resulting from collaborations between my team and biologists. For instance, SVJedi-graph has been used on polyploid genomes of plant pathogenic nematodes (collaboration with Etienne

[1] https://github.com/SandraLouise/SVJedi-graph
[2] https://anaconda.org/bioconda/svjedi-graph

Danchin from Inrae, Nice). In this project, several samples were sequenced with both short and long reads and SVJedi-graph was used to validate with long read the SV callsets obtained with the short ones. One of the feedback I got from this analysis was the confirmation that raw SV callsets obtained with short read SV callers are really messy and contains many close or overlapping SV calls. In another collaboration (with Claire Mérot from the ecological lab Ecobio, Rennes), SVJedi-graph is currently being used on the public data from the Darwin Tree Of Life project[3]. This British project, led by the Sanger Institute, is one of the largest sequencing projects, and aims to sequence and analyse the genomes of more than 70,000 living organisms. In this collaboration, the aim is to analyze heterozygous SVs from various diploid organisms. Each organism has been sequenced with the same standardized protocol with the PacBio HiFi technology. A pipeline of SV calling has been developed and SVJedi-graph is integrated as one of the last steps, after SV calling by several SV callers, in order to filter out SV calls that do not have the expected heterozygous genotype.

Managing imprecise breakpoints

The main identified limit of SVJedi-graph is its sensitiveness to breakpoint imprecision. I got confronted to this limit when attempting to genotype the large inversions discovered in the *Coenonympha* genomes, for which I estimated a median breakpoint imprecision of 7 kbp, that even reached 50 kbp for one of the inversion's breakpoint. When SVs are represented with inaccurate breakpoint positions in the graph, the reads supporting the alternative allele are most likely to be mapped in several parts (*i.e.* split-mappings), as the edge in the graph representing the alternative sequence adjacency is erroneous. At the moment, SVJedi-graph analyzes all the alignments of one read independently. As a consequence, in the particular case of inversions, a read supporting the inversion will be interpreted as a support for the reference allele, as no alignment will overlap the SV breakpoints. In order to better handle imprecise breakpoints, the method could be improved to take into account split mappings. For each read, a chaining of its split mappings could be performed to identify which of the allele paths of the graph fits best to the whole read sequence.

Furthermore, these split mappings could pinpoint SVs for which the breakpoints given in the input VCF are imprecise, and so be used to warn the user of this imprecision in the output of SVJedi-graph. If the read depth is sufficient enough, we could even consider analyzing the positions on the graph where the read mapping split happen (*e.g.* through clustering of these positions between the reads) and submit corrected breakpoint positions to the user. Validation tests

[3] https://www.sanger.ac.uk/collaboration/darwin-tree-of-life-project/

could be performed by constructing a subgraph of the SV with the proposed corrected breakpoints and checking that the formerly split reads do not produce split mappings on the corrected subgraph.

Adaptation of SVJedi-graph's method to pangenome graphs

SVJedi-graph's method could also be adapted to genotype SVs on pangenome graphs, in a similar fashion as what is done by Giraffe, which would allow the user to bypass the constraints of establishing qualitative SV sets. The graph construction step would be skipped, the pangenome graph taken as input. The input VCF would be one describing the bubbles of the pangenome graph (*e.g.* a VCF obtained from *vg deconstruct*), with some filters on the bubble's size. The edges representing each SV's allele would be deduced from the VCF. Very little to no modification would be needed from the read counting and genotype inference steps of SVJedi-graph. A crucial question would be the choice of the read mapper to use, as pangenome graphs are much more complex graphs (*i.e.* bigger size, node and edge number) than the variation graphs of SVJedi-graph. In particular, we can expect small nested variants to be represented in SV bubbles. This could pose a problem for the minigraph mapper, which fails to map reads (or any sequences) on subgraphs consiting of many small successive nodes. If such subgraph happens to be in close proximity of a SV breakpoint, the genotyping rate would drop. However, due to their higher graph complexity, the question of whether pangenome graphs are better than variation graph to genotype SVs is still open.

Additionally, SVJedi-graph's genotyping method applied to pangenome graph would once again be susceptible to breakpoint imprecision, which was observed to happen in pangenome graphs already elsewhere. For instance, Audano and Beck (2024) quantified the proportion of SVs with imprecise breakpoints positions to reach 69 and 41 % for deletions and insertions in human pangenome graphs constructed by Minigraph-Cactus. I also observed breakpoint imprecision in pangenome graphs in the specific case of inversions.

Uncompleteness of structural variant genotyping evaluation

Evaluating the performances of SV genotyping methods on realistic data requires datasets with SV sets that are accurately described both on their position and their genotype in sequenced samples. Such datasets require sizeable efforts to construct, which make them precious to the community and explains why they are few and mainly focused on human data. Despite their quality, they do not represent the whole range of SV types. For instance, the high-quality benchmark dataset established by the Genome in a Bottle consortium for the human genome (Zook

et al., 2020) exclusively contains deletions and insertions in a simplified genomic context (*i.e.* without SV proximity). This dataset has been extensively used for benchmarking SV genotyping tools, but is not representative of more realistic SV sets obtained with SV discovery tools, which often contain close SVs. This limits the evaluation of genotyping performances when developing new methods. In the past few years, new more diverse datasets were made available, such as a dataset of challenging SVs from GIAB and a high quality dataset of human inversions from Porubsky et al. (2022), and hopefully such high quality datasets will keep getting more diversified in the future.

Inversion discovery: reference genome *versus* pangenome graph

In the first phase of genome comparison between the *Coenonympha* species, whole-genome alignment plots (dot-plots) between the two parental species genomes confirmed the presence of several large inversions. In order to better characterize these inversions (*i.e.* identifying more precise breakpoint positions) and discover the whole panel of large inversions in the complex of species, I tested two inversion discovery strategies, which I discuss in the two following sections.

Reference-based SV discovery tools

In my preliminary work to find the most fitting approach for our data, I tested six state of the art tools covering the two main approaches: long-read mapping (Sniffles2, SVIM, cuteSV) and assembly alignment (SVIM-asm, MUM&Co, SyRI). At the exception of SyRI, none of the tools were able to identify more than one of the large inversions observed on the alignment plots. These results highlight the fact that SV discovery tools are primarily designed for intra-species comparison, especially read mapping-based ones. Their method is tuned for low levels of divergence in the alignments, and was initially evaluated on human data. On the other hand, I also tested inter-specific WGA tools (such as AnchorWave), but they are not designed to output a list of variants or precise breakpoints. The better results I obtained with SyRI might result from its method based on the search for synteny blocks, which could render it relatively more lenient towards genomic divergence. Nonetheless, I was able to produce a list of twelve high-confidence large inversions differentiating the genomes of the parental species, through manual curation of the results obtained from SyRI.

Although SyRI achieved a good characterization of the large inversions between the *Coenonympha* genomes, the tool requires both compared assemblies to be

of chromosome-level contiguity (*i.e.* with an identical number of chromosomes). Such contiguity can be obtained with scaffolding tools such as RagTag (Alonge et al., 2022) or chroder Goel et al. (2019), but the scaffolding process might lead to the loss of rearrangements in assemblies. If a contig comprises a rearrangement on the majority of its length, the scaffolding process will reorder and reorient it to match the reference genome. In fact, an instance of this issue occurred when characterizing the large *Coenonympha* inversions with SyRI (mentioned in Paper 1). In parallel to the test of SV discovery tools, I developed a method to characterize large inversions based on synteny blocks analysis without such requirements. The method compares the sequence of synteny blocks along two compared chromosomes, and finds breakpoint pairs respecting the typical properties of inversions. I tested this method using synteny blocks produced by SibeliaZ on the *Coenonympha* genomes, and obtained a very similar large inversion recall as that of SyRI. This work remains as a draft method, nevertheless promising for the identification of large inversions when genome assemblies are fragmented or when the compared genomes contain different numbers of chromosomes. The approximative breakpoint positions of the identified inversions, resulting from the gaps between synteny blocks, could be refined through local sensitive alignment, using Cassis (Baudet et al., 2010) for example. It could even be extended to other types of large rearrangements (*e.g.* translocations, deletions).

A drawback of the reference-based approach remains that they require to perform n pairwise genome comparisons, after which the SV calls need to be merged.

Inversion annotation from pangenome graphs

The second strategy I tested was inversion discovery from pangenome graphs. This strategy appears the most intuitive in the *Coenonympha* application case, where designating a parental genome as a reference is not needed. The problem I encountered was the identification of inversions among the graph bubbles, that is differentiating bubbles representing inversions from bubbles representing other types of variants. To resolve this problem, I developed a method to annotate inversions among the graph bubbles reported by state of the art tools *vg deconstruct* and *gfatools bubble*. This method is able to annotate inversions that are explicitly represented by the paths of bubbles, as well as inversions that are represented as unrelated sequences in the bubbles.

I demonstrated that the various pipelines used to construct pangenome graphs represent inversions in different ways. More importantly, I showed that the inversion discovery recall on pangenome graphs could be highly uneven depending on the constructing pipeline used, and overall surprisingly low even for low-complexity

simulated data. The relatively low stringency of the method would lead me to suggest that the missed inversions are either not represented in a single bubbles (*i.e.* represented as a chain of bubbles) or their bubble might not be recognized as variant bubbles by *vg deconstruct* or *gfatools bubble*. This later hypothesis comes from the observation that in some cases, a few bubbles reported by one tool are not reported by the other tool. A final hypothesis would be that some inversions are not represented altogether in the graph's topography.

A way to test these hypotheses, and hopefully improve the annotation rate, would be to annotate inversions without use of an intermediary bubble detection tools, by directly looking for inversion signals in the graph. The detection of inversions represented as unrelated sequences in the graph could also be a first step towards a correction of pangenome graphs, as such occurrences add complexity and noise to the graphs. They could for example hinder the analysis and genotyping of small variants nested in these inversions.

Final thoughts on inter-specific inversion discovery

Among the various strategies tested for inversion discovery across multiple genomes of different species, none showed a clear superiority over the others. On one hand, the reference-based strategy provides more well tested tool options to choose from and in the end yielded better results. On the other hand, the pangenome graph-based strategy is more suited when comparing more than two genomes, returning a list of variants that can be compared simultaneously between all genomes, and can be used without having to designate a reference genome. However, there is at the moment a lack of tools to correctly identify, analyze and exploit large variants in pangenome graphs.

Bibliography

Ahsan, M. U., Liu, Q., Perdomo, J. E., Fang, L., and Wang, K. (2023). A survey of algorithms for the detection of genomic structural variants from long-read sequencing data. *Nature Methods*, 20(8):1143–1158.

Alonge, M., Lebeigle, L., Kirsche, M., Jenike, K., Ou, S., Aganezov, S., Wang, X., Lippman, Z. B., Schatz, M. C., and Soyk, S. (2022). Automated assembly scaffolding using ragtag elevates a new tomato system for high-throughput genome editing. *Genome biology*, 23(1):258.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.

Armstrong, J., Fiddes, I. T., Diekhans, M., and Paten, B. (2019). Whole-genome alignment and comparative annotation. *Annual review of animal biosciences*, 7(1):41–64.

Armstrong, J., Hickey, G., Diekhans, M., Fiddes, I. T., Novak, A. M., Deran, A., Fang, Q., Xie, D., Feng, S., Stiller, J., et al. (2020). Progressive cactus is a multiple-genome aligner for the thousand-genome era. *Nature*, 587(7833):246–251.

Audano, P. A. and Beck, C. R. (2024). Small polymorphisms are a source of ancestral bias in structural variant breakpoint placement. *Genome research*, 34(1):7–19.

Audano, P. A., Sulovari, A., Graves-Lindsay, T. A., Cantsilieris, S., Sorensen, M., Welch, A. E., Dougherty, M. L., Nelson, B. J., Shah, A., Dutcher, S. K., et al. (2019). Characterizing the major structural variant alleles of the human genome. *Cell*, 176(3):663–675.

Baudet, C., Lemaitre, C., Dias, Z., Gautier, C., Tannier, E., and Sagot, M.-F. (2010). Cassis: detection of genomic rearrangement breakpoints. *Bioinformatics*, 26(15):1897–1898.

Bayer, P. E., Petereit, J., Durant, É., Monat, C., Rouard, M., Hu, H., Chapman, B., Li, C., Cheng, S., Batley, J., et al. (2022). Wheat panache: A pangenome graph database representing presence–absence variation across sixteen bread wheat genomes. *The Plant Genome*, 15(3):e20221.

Belyayev, A. (2014). Bursts of transposable elements as an evolutionary driving force. *Journal of evolutionary biology*, 27(12):2573–2584.

Berdan, E. L., Barton, N. H., Butlin, R., Charlesworth, B., Faria, R., Fragata, I., Gilbert, K. J., Jay, P., Kapun, M., Lotterhos, K. E., et al. (2023). How chromosomal inversions reorient the evolutionary process. *Journal of evolutionary biology*, 36(12):1761–1782.

Beyter, D., Ingimundardottir, H., Oddsson, A., Eggertsson, H. P., Bjornsson, E., Jonsson, H., Atlason, B. A., Kristmundsdottir, S., Mehringer, S., Hardarson, M. T., et al. (2021). Long-read sequencing of 3,622 icelanders provides insight into the role of structural variants in human diseases and other traits. *Nature genetics*, 53(6):779–786.

Byrska-Bishop, M., Evani, U. S., Zhao, X., Basile, A. O., Abel, H. J., Regier, A. A., Corvelo, A., Clarke, W. E., Musunuri, R., Nagulapalli, K., et al. (2022). High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440.

Capblancq, T., Després, L., and Mavárez, J. (2020). Genetic, morphological and ecological variation across a sharp hybrid zone between two alpine butterfly species. *Evolutionary applications*, 13(6):1435–1450.

Capblancq, T., Després, L., Rioux, D., and Mavárez, J. (2015). Hybridization promotes speciation in coenonympha butterflies. *Molecular Ecology*, 24(24):6209–6222.

Capblancq, T., Mavárez, J., Rioux, D., and Després, L. (2019). Speciation with gene flow: evidence from a complex of alpine butterflies (coenonympha, satyridae). *Ecology and evolution*, 9(11):6444–6457.

Chaisson, M. J., Sanders, A. D., Zhao, X., Malhotra, A., Porubsky, D., Rausch, T., Gardner, E. J., Rodriguez, O. L., Guo, L., Collins, R. L., et al. (2019). Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nature communications*, 10(1):1784.

Chen, S., Krusche, P., Dolzhenko, E., Sherman, R. M., Petrovski, R., Schlesinger, F., Kirsche, M., Bentley, D. R., Schatz, M. C., Sedlazeck, F. J., et al. (2019). Paragraph: a graph-based structural variant genotyper for short-read sequence data. *Genome biology*, 20:1–13.

Cheng, H., Concepcion, G. T., Feng, X., Zhang, H., and Li, H. (2021). Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature methods*, 18(2):170–175.

Cochetel, N., Minio, A., Guarracino, A., Garcia, J. F., Figueroa-Balderas, R., Massonnet, M., Kasuga, T., Londo, J. P., Garrison, E., Gaut, B. S., et al. (2023).

A super-pangenome of the north american wild grape species. *Genome Biology*, 24(1):290.

Collins, R. L., Brand, H., Karczewski, K. J., Zhao, X., Alföldi, J., Francioli, L. C., Khera, A. V., Lowther, C., Gauthier, L. D., Wang, H., et al. (2020). A structural variation reference for medical and population genetics. *Nature*, 581(7809):444–451.

Dabbaghie, F., Ebler, J., and Marschall, T. (2022). Bubblegun: enumerating bubbles and superbubbles in genome graphs. *Bioinformatics*, 38(17):4217–4219.

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. (2011). The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158.

Darling, A. C., Mau, B., Blattner, F. R., and Perna, N. T. (2004). Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome research*, 14(7):1394–1403.

Delage, W. J., Thevenon, J., and Lemaitre, C. (2020). Towards a better understanding of the low recall of insertion variants with short-read based variant callers. *BMC genomics*, 21:1–17.

Denti, L., Khorsand, P., Bonizzoni, P., Hormozdiari, F., and Chikhi, R. (2023). Svdss: structural variation discovery in hard-to-call genomic regions using sample-specific strings from accurate long reads. *Nature Methods*, 20(4):550–558.

Després, L., Henniaux, C., Rioux, D., Capblancq, T., Zupan, S., Čelik, T., Sielezniew, M., Bonato, L., and Ficetola, G. F. (2019). Inferring the biogeography and demographic history of an endangered butterfly in europe from multilocus markers. *Biological Journal of the Linnean Society*, 126(1):95–113.

Döring, A., Weese, D., Rausch, T., and Reinert, K. (2008). Seqan an efficient, generic c++ library for sequence analysis. *BMC bioinformatics*, 9:1–9.

D'Costa, A. V. and Simpson, J. T. (2023). Somrit: The somatic retrotransposon insertion toolkit. *bioRxiv*, pages 2023–08.

Ebler, J., Ebert, P., Clarke, W. E., Rausch, T., Audano, P. A., Houwaart, T., Mao, Y., Korbel, J. O., Eichler, E. E., Zody, M. C., et al. (2022). Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. *Nature genetics*, 54(4):518–525.

Eggertsson, H. P., Kristmundsdottir, S., Beyter, D., Jonsson, H., Skuladottir, A., Hardarson, M. T., Gudbjartsson, D. F., Stefansson, K., Halldorsson, B. V., and Melsted, P. (2019). Graphtyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nature communications*, 10(1):5402.

Garrison, E. and Guarracino, A. (2023). Unbiased pangenome graphs. *Bioinformatics*, 39(1):btac743.

Garrison, E., Guarracino, A., Heumos, S., Villani, F., Bao, Z., Tattini, L., Hagmann, J., Vorbrugg, S., Marco-Sola, S., Kubica, C., et al. (2023). Building pangenome graphs. *bioRxiv*, pages 2023–04.

Garrison, E., Kronenberg, Z. N., Dawson, E. T., Pedersen, B. S., and Prins, P. (2022). A spectrum of free software tools for processing the vcf variant call format: vcflib, bio-vcf, cyvcf2, hts-nim and slivar. *PLOS Computational Biology*, 18:1–15.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., et al. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36(9):875–879.

Goel, M., Sun, H., Jiao, W.-B., and Schneeberger, K. (2019). Syri: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome biology*, 20:1–13.

Heller, D. and Vingron, M. (2019). Svim: structural variant identification using mapped long reads. *Bioinformatics*, 35(17):2907–2915.

Heller, D. and Vingron, M. (2020). Svim-asm: structural variant detection from haploid and diploid genome assemblies. *Bioinformatics*, 36(22-23):5519–5521.

Hickey, G., Monlong, J., Ebler, J., Novak, A. M., Eizenga, J. M., Gao, Y., Marschall, T., Li, H., and Paten, B. (2024). Pangenome graph construction from genome alignments with minigraph-cactus. *Nature biotechnology*, 42(4):663–673.

Hidalgo, O., Pellicer, J., Christenhusz, M., Schneider, H., Leitch, A. R., and Leitch, I. J. (2017). Is there an upper limit to genome size? *Trends in Plant Science*, 22(7):567–573.

Huang, K. and Rieseberg, L. H. (2020). Frequency, origins, and evolutionary role of chromosomal inversions in plants. *Frontiers in plant science*, 11:296.

Jain, C., Rhie, A., Hansen, N. F., Koren, S., and Phillippy, A. M. (2022). Long-read mapping to repetitive reference sequences using winnowmap2. *Nature Methods*, 19(6):705–710.

Jang, J., Jung, J., Lee, Y. H., Lee, S., Baik, M., and Kim, H. (2023). Chromosome-level genome assembly of korean native cattle and pangenome graph of 14 bos taurus assemblies. *Scientific Data*, 10(1):560.

Jiang, T., Liu, S., Cao, S., and Wang, Y. (2022). Structural variant detection from long-read sequencing data with cutesv. In *Variant Calling: Methods and Protocols*, pages 137–151. Springer.

Joron, M., Frezal, L., Jones, R. T., Chamberlain, N. L., Lee, S. F., Haag, C. R., Whibley, A., Becuwe, M., Baxter, S. W., Ferguson, L., et al. (2011). Chromosomal rearrangements maintain a polymorphic supergene controlling butterfly mimicry. *Nature*, 477(7363):203–206.

Kebaïli, C., Sherpa, S., Gueguen, M., Renaud, J., Rioux, D., and Després, L. (2023). Comparative genetic and demographic responses to climate change in three peatland butterflies in the jura massif. *Biological Conservation*, 287:110332.

Kent, W. J. (2002). Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664.

Kiełbasa, S. M., Wan, R., Sato, K., Horton, P., and Frith, M. C. (2011). Adaptive seeds tame genomic sequence comparison. *Genome research*, 21(3):487–493.

Kirkpatrick, M. (2010). How and why chromosome inversions evolve. *PLoS biology*, 8(9):e1000501.

Kolmogorov, M., Armstrong, J., Raney, B. J., Streeter, I., Dunn, M., Yang, F., Odom, D., Flicek, P., Keane, T. M., Thybert, D., et al. (2018). Chromosome assembly of large and complex genomes using multiple references. *Genome research*, 28(11):1720–1732.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., and Phillippy, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research*, 27(5):722–736.

Lecompte, L., Peterlongo, P., Lavenier, D., and Lemaitre, C. (2020). Svjedi: genotyping structural variations with long reads. *Bioinformatics*, 36(17):4568–4575.

Legeai, F., Romain, S., Capblancq, T., Doniol-Valcroze, P., Joron, M., Lemaitre, C., and Després, L. (2024). Chromosome-level assembly and annotation of the pearly heath coenonympha arcania butterfly genome. *Genome Biology and Evolution*, 16(3):evae055.

Leonard, A. S., Crysnanto, D., Mapel, X. M., Bhati, M., and Pausch, H. (2023). Graph construction method impacts variation representation and analyses in a bovine super-pangenome. *Genome Biology*, 24(1):124.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100.

Li, H. (2019). gfatools. https://github.com/lh3/gfatools.

Li, H. and Durbin, R. (2024). Genome assembly in the telomere-to-telomere era. *Nature Reviews Genetics*, pages 1–13.

Li, H., Feng, X., and Chu, C. (2020). The design and construction of reference pangenome graphs with minigraph. *Genome biology*, 21:1–19.

Liao, W.-W., Asri, M., Ebler, J., Doerr, D., Haukness, M., Hickey, G., Lu, S., Lucas, J. K., Monlong, J., Abel, H. J., et al. (2023). A draft human pangenome reference. *Nature*, 617(7960):312–324.

Lin, H.-N. and Hsu, W.-L. (2020). Gsalign: an efficient sequence alignment tool for intra-species genomes. *BMC genomics*, 21:1–10.

Lin, J., Wang, S., Audano, P. A., Meng, D., Flores, J. I., Kosters, W., Yang, X., Jia, P., Marschall, T., Beck, C. R., et al. (2022). Svision: a deep learning approach to resolve complex structural variants. *Nature methods*, 19(10):1230–1233.

Liu, Z., Xie, Z., and Li, M. (2024). Comprehensive and deep evaluation of structural variation detection pipelines with third-generation sequencing data. *Genome Biology*, 25(1):188.

Lohse, K., of Life Project Consortium, D. T., et al. (2021a). The genome sequence of the speckled wood butterfly, pararge aegeria (linnaeus, 1758). *Wellcome Open Research*, 6(287).

Lohse, K., Weir, J., of Life, W. S. I. T., of Life Consortium, D. T., et al. (2021b). The genome sequence of the meadow brown, maniola jurtina (linnaeus, 1758). *Wellcome Open Research*, 6.

Mahmoud, M., Gobet, N., Cruz-Dávalos, D. I., Mounier, N., Dessimoz, C., and Sedlazeck, F. J. (2019). Structural variant calling: the long and the short of it. *Genome biology*, 20:1–14.

Marçais, G., Delcher, A. L., Phillippy, A. M., Coston, R., Salzberg, S. L., and Zimin, A. (2018). Mummer4: A fast and versatile genome alignment system. *PLoS computational biology*, 14(1):e1005944.

Miao, J., Wei, X., Cao, C., Sun, J., Xu, Y., Zhang, Z., Wang, Q., Pan, Y., and Wang, Z. (2024). Pig pangenome graph reveals functional features of non-reference sequences. *Journal of Animal Science and Biotechnology*, 15(1):32.

Minkin, I. and Medvedev, P. (2020). Scalable multiple whole-genome alignment and locally collinear block construction with sibeliaz. *Nature communications*, 11(1):6327.

Mirus, T., Lohmayer, R., Halldorsson, B. V., and Kehr, B. (2024). Ggtyper: genotyping complex structural variants using short-read sequencing data. *bioRxiv*, pages 2024–03.

Nattestad, M. and Schatz, M. C. (2016). Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics*, 32(19):3021–3023.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

O'donnell, S. and Fischer, G. (2020). Mum&co: accurate detection of all sv types through whole-genome alignment. *Bioinformatics*, 36(10):3242–3243.

Paten, B., Earl, D., Nguyen, N., Diekhans, M., Zerbino, D., and Haussler, D. (2011). Cactus: Algorithms for genome multiple sequence alignment. *Genome research*, 21(9):1512–1528.

Paten, B., Novak, A. M., Eizenga, J. M., and Garrison, E. (2017). Genome graphs and the evolution of genome inference. *Genome research*, 27(5):665–676.

Peng, Q., Alekseyev, M. A., Tesler, G., and Pevzner, P. A. (2009). Decoding synteny blocks and large-scale duplications in mammalian and plant genomes. In *Algorithms in Bioinformatics: 9th International Workshop, WABI 2009, Philadelphia, PA, USA, September 12-13, 2009. Proceedings 9*, pages 220–232. Springer.

Porubsky, D., Höps, W., Ashraf, H., Hsieh, P., Rodriguez-Martin, B., Yilmaz, F., Ebler, J., Hallast, P., Maggiolini, F. A. M., Harvey, W. T., et al. (2022). Recurrent inversion polymorphisms in humans associate with genetic instability and genomic disorders. *Cell*, 185(11):1986–2005.

Rautiainen, M. and Marschall, T. (2020). Graphaligner: rapid and versatile sequence-to-graph alignment. *Genome biology*, 21(1):253.

Rice, E. S., Alberdi, A., Alfieri, J., Athrey, G., Balacco, J. R., Bardou, P., Blackmon, H., Charles, M., Cheng, H. H., Fedrigo, O., et al. (2023). A pangenome graph reference of 30 chicken genomes allows genotyping of large and complex structural variants. *BMC biology*, 21(1):267.

Romain, S., Capblancq, T., Despres, L., Joron, M., Legeai, F., and Lemaitre, C. (in prep.a). Characterization of large inversions to investigate hybrid speciation in the four species-complex of alpine coenonympha butterfly.

Romain, S., Dubois, S., Legeai, F., and Lemaitre, C. (in prep.b). Investigating the topological motifs of inversions in pangenome graphs.

Romain, S. and Lemaitre, C. (2023). Svjedi-graph: improving the genotyping of close and overlapping structural variants with long reads using a variation graph. *Bioinformatics*, 39(Supplement_1):i270–i278.

Sahlin, K., Baudeau, T., Cazaux, B., and Marchet, C. (2023). A survey of mapping algorithms in the long-reads era. *Genome Biology*, 24(1):133.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467.

Sedlazeck, F. J., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., Von Haeseler, A., and Schatz, M. C. (2018). Accurate detection of complex structural variations using single-molecule sequencing. *Nature methods*, 15(6):461–468.

Sherpa, S., Kebaïli, C., Rioux, D., Guéguen, M., Renaud, J., and Després, L. (2022). Population decline at distribution margins: assessing extinction risk in the last glacial relictual but still functional metapopulation of a european butterfly. *Diversity and Distributions*, 28(2):271–290.

Sirén, J., Monlong, J., Chang, X., Novak, A. M., Eizenga, J. M., Markello, C., Sibbesen, J. A., Hickey, G., Chang, P.-C., Carroll, A., et al. (2021). Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science*, 374(6574):abg8871.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

Smolka, M., Paulin, L. F., Grochowski, C. M., Horner, D. W., Mahmoud, M., Behera, S., Kalef-Ezra, E., Gandhi, M., Hong, K., Pehlivan, D., et al. (2024). Detection of mosaic and population-level structural variants with sniffles2. *Nature biotechnology*, pages 1–10.

Song, B., Marco-Sola, S., Moreto, M., Johnson, L., Buckler, E. S., and Stitzer, M. C. (2022). Anchorwave: Sensitive alignment of genomes with high sequence diversity, extensive structural polymorphism, and whole-genome duplication. *Proceedings of the National Academy of Sciences*, 119(1):e2113075119.

Taylor, D. J., Eizenga, J. M., Li, Q., Das, A., Jenike, K. M., Kenny, E. E., Miga, K. H., Monlong, J., McCoy, R. C., Paten, B., et al. (2024). Beyond the human genome project: The age of complete human genome sequences and pangenome references. *Annual Review of Genomics and Human Genetics*, 25.

Thompson, M. J. and Jiggins, C. (2014). Supergenes and their role in evolution. *Heredity*, 113(1):1–8.

Trevors, J. (1996). Genome size in bacteria. *Antonie van Leeuwenhoek*, 69:293–303.

Wellenreuther, M. and Bernatchez, L. (2018). Eco-evolutionary genomics of chromosomal inversions. *Trends in ecology & evolution*, 33(6):427–440.

Zhao, X., Collins, R. L., Lee, W.-P., Weber, A. M., Jun, Y., Zhu, Q., Weisburd, B., Huang, Y., Audano, P. A., Wang, H., et al. (2021). Expectations and blind spots for structural variation detection from long-read assemblies and short-read genome sequencing technologies. *The American Journal of Human Genetics*, 108(5):919–928.

Zhao, X., Weber, A. M., and Mills, R. E. (2017). A recurrence-based approach for validating structural variation using long-read sequencing technology. *Gigascience*, 6(8):gix061.

Zook, J. M., Hansen, N. F., Olson, N. D., Chapman, L., Mullikin, J. C., Xiao, C., Sherry, S., Koren, S., Phillippy, A. M., Boutros, P. C., et al. (2020). A robust benchmark for detection of germline large deletions and insertions. *Nature biotechnology*, 38(11):1347–1355.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Identification, génotypage et représentation des variants structuraux dans les pangénomes

Mots clés : variants structuraux – inversions – génotypage – identification d'inversions – graphes de variation – graphes de pangénomes

Résumé : Les variants structuraux (SVs), des variations génomiques de plus de 50 pb, contribuent de manière significative à la diversité génétique et à l'évolution des espèces. La détection et le génotypage précis des SVs est crucial pour comprendre leur rôle dans la variation phénotypique et l'adaptation. Les graphes de variation (VGs) et graphes de pangénomes (PGs), qui représentent les variations génomiques comme des chemins alternatifs dans un graphe, offrent une approche prometteuse pour l'analyse des SVs. Cette thèse explore l'utilisation des VGs et PGs pour la détection et le génotypage des SVs, en se concentrant sur un complexe de quatre espèces de papillons *Coenonympha* alpins. Deux outils bio-informatiques ont été développés au cours de cette thèse : (1) SVJedi-graph, le premier génotypeur de SVs à partir de lectures longues utilisant un VG pour représenter les SVs, fournissant une précision de génotypage supérieure aux outils de l'état de l'art, en particulier pour les SVs proches et chevauchants, et (2) INVPG-annot, un outil d'identification des inversions dans les PGs, qui a permis de démontrer que les inversions sont représentées par différentes topologies dans les PGs selon l'outil de construction utilisé. L'analyse comparative des génomes des papillons *Coenonympha* a permis d'identifier douze grandes inversions (≥ 100 kbp) entre les quatre espèces, dont certaines pourraient jouer un rôle dans l'isolement reproducteur et l'adaptation locale de deux de ces espèces. Bien que l'approche basée sur les PGs présente des avantages pour la comparaison de génomes, des défis restent à relever pour l'analyse des grands variants comme les inversions.

Title: Identification, genotyping and representation of structural variants in pangenomes

Keywords: structural variants – inversions – genotyping – inversion identification – variation graph – pangenome graph

Abstract: Structural variants (SVs), genomic variations of more than 50 bp, contribute significantly to genetic diversity and species evolution. Accurate detection and genotyping SVs is crucial to understanding their role in phenotypic variation and adaptation. Variation graphs (VGs) and pangenome graphs (PGs), which represent genomic variations as alternative paths in a graph, offer a promising approach for the analysis of SVs. This thesis explores the use of VGs and PGs for the detection and genotyping of SVs, focusing on a complex of four species of alpine *Coenonympha* butterflies. Two bioinformatics tools were developed during this thesis: (1) SVJedi-graph, the first long-read SV genotyper using a VG to represent SVs, providing a genotyping accuracy superior to state-of-the-art tools, particularly for close and overlapping SVs, and (2) INVPG-annot, a tool for identifying inversions in PGs, which demonstrated that inversions are represented by different topologies in PGs depending on the construction tool used. Comparative analysis of the *Coenonympha* butterfly genomes identified twelve large inversions (≥ 100 kbp) between the four species, some of which could play a role in the reproductive isolation and local adaptation of two of these species. While the PG-based approach offers advantages for genome comparison, challenges remain for the analysis of large variants such as inversions.