On the origin of ‘bloopergenes’: unraveling the evolution of the balanced lethal system in Triturus newts

Visser, M.C. de

Triturus *embryo breeding, Belgrade University, Serbia*

© Manon de Visser

Chapter 5 - PAV-spotter: using signal cross-correlations to identify Presence/Absence Variation in target capture data

Manon de Visser[1,2], Chris van der Ploeg[3,4], Milena Cvijanović[5], Tijana Vučić[1,2,6]*,* Anagnostis Theodoropoulos[1,2], Ben Wielstra[1,2]

1. Institute of Biology Leiden, Faculty of Science, Leiden University, Leiden, The Netherlands
2. Naturalis Biodiversity Center, Leiden, The Netherlands
3. Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
4. Integrated Vehicle Safety Group, Netherlands Organisation for Applied Scientific Research, Helmond, The Netherlands
5. Institute for Biological Research "Siniša Stanković", National Institute of the Republic of Serbia, Department of Evolutionary Biology, University of Belgrade, Belgrade, Serbia
6. Institute of Zoology, Faculty of Biology, University of Belgrade, Belgrade, Serbia

This manuscript has been submitted to the journal ***Molecular Ecology Resources*** and is currently under review. A preprinted version of this manuscript has been published at ***bioRxiv*** (2024.10.25.620064; https://doi.org/10.1101/2024.10.25.620064).

Abstract

High throughput sequencing technologies have become essential in the fields of evolutionary biology and genomics. When dealing with non-model organisms or genomic gigantism, sequencing whole genomes is still relatively costly and therefore reduced-genome representations are frequently obtained, for instance by 'target capture' approaches. While computational tools exist that can handle target capture data and identify small-scale variants such as single nucleotide polymorphisms and micro-indels, options to identify large scale structural variants are limited. To meet this need, we introduce PAV-spotter: a tool that can identify presence/absence variation (PAV) in target capture data. PAV-spotter conducts a signal cross-correlation calculation, in which the distribution of read counts per target between samples of different *a priori* defined classes – e.g. male versus female, or diseased versus healthy – are compared. We apply and test our methodology by studying *Triturus* newts: salamanders with gigantic genomes that currently lack an annotated reference genome. *Triturus* newts suffer from a hereditary disease that kills half their offspring during embryogenesis. We compare the target capture data of two different types of diseased embryos, characterized by unique deletions, with those of healthy embryos. Our findings show that PAV-spotter helps to expose such structural variants, even in the face of medium to low sequencing coverage levels, low sample sizes, and background noise due to mis-mapped reads. PAV-spotter can be used to study the structural variation underlying supergene systems in the absence of whole genome assemblies. The code, including further explanation, is available through the PAV-spotter GitHub repository: https://github.com/Wielstra-Lab/PAVspotter.

Keywords

PA variation, Hyb-Seq , target enrichment, copy number variation, pattern recognition, *Triturus*, supergenes, hemizygosity.

Introduction

Next-generation DNA sequencing methods have revolutionized the biological sciences, with an ever-growing amount of sequence data being generated worldwide [1, 2]. High throughput sequencing techniques have become more affordable and increasingly used, however sequencing entire genomes can still be challenging, for instance when dealing with non-model organisms, genomic gigantism, or a combination of the two [3-5]. In such cases, well-annotated reference genomes for aligning (re-)sequenced reads are generally unavailable, making whole genome sequencing relatively costly in terms of money and (computational) time [e.g. see; 6, 7, 8]. Many biologists therefore still opt for more cheap and efficient 'reduced-representation' high throughput sequencing techniques, which allow for a subset of loci to be sequenced more deeply [9-11].

A technique that has become particularly popular for studying non-model species is target capture, also referred to as hybridization sequencing, exon capture, sequence capture, or exome capture [11-14]. This method facilitates collecting sequence information on hundreds or thousands of pre-selected loci. Due to the consequent rise of large multi-locus DNA sequence datasets, the need for innovative, easy-to-implement bioinformatic applications has surged as well [15]. User-friendly pipelines help to pre-process sequence reads by wrapping and connecting existing software tools, with well-known examples for target capture including HybPiper [16], Assexon [17] and Sequence Capture Processor [i.e. SECAPR, see; 18]. These pipelines and their software dependencies support fast upstream data cleaning and guide the user to phylogenetic applications downstream.

Typically, target capture analysis pipelines are utilized to identify small-scale genetic variants such as single nucleotide polymorphisms (SNPs) and relatively small insertions/deletions (microindels, ranging from 1~50bp) to perform, for instance, phylogenetic tree-building [10]. However, to identify and analyze larger scale information from target capture data, such as genomic structural variation, few tools are available – especially when focusing on non-model organisms. A particular type of larger-scale variation that is hard to identify in multi-locus DNA sequence datasets is presence/absence variation (hereafter referred to as 'PAV') of relatively big insertions/ deletions (macroindels, > 50bp), i.e. above the size of microindels [19, 20].

Structural variants such as PAVs are regularly overlooked because they are harder to identify than SNPs [21, 22]. However they are a major source of genetic divergence and diversity [23-26]. PAV in particular poses an extreme example of copy number variation, where fragments in the size of entire exons or (stretches of DNA containing multiple) genes are missing from one genome compared to another [27, 28]. When comparing such genomes, target capture data would in theory display PAV by showing ‘normal data’ in the case of target presence, versus a ‘data gap’ in the case of target absence. This would be an indication of structural variation.

Whether there are consistent differences in presence or absence of sequence data can be determined by analyzing the way that reads pile up against a certain reference set of sequences. Some tools can detect copy number variation and PAV patterns by comparing the depth of mapped reads of different samples, such as ExomeCNV (Sathirapongsasuti et al, 2011) and SUPER-CAP (Yuan et al, 2019). However, these tools come with strict requirements, such as good quality reference genomes being available, known functional annotation of variants, and/or coverage being consistently high across all samples and targets, with mapped reads ideally following a normal distribution. Yet, most multi-locus datasets, including those resulting from target capture experiments, generally do not meet such requirements [10].

We introduce PAV-spotter: a flexible signal cross-correlation method that is able to ‘spot’ potential PAV in target capture datasets. Our approach borrows the notion of cross-correlation to detect the dissimilarity between datasets obtained through target capture experiments. Cross-correlation methods are generally used in the domain of control engineering. Classically, they are applied on time-series data [29], for example in machinery fault detection studies [30]. However, cross-correlation approaches have been proven useful in the field of pattern recognition as well [31].

Being able to identify structural variation by using pattern recognition would be especially informative when studying supergenes systems, in which individuals can have zero, one or two copies of particular loci [32, 33]. Supergenes consist of genes that are inherited together as a single locus due to the suppression of recombination [33-35]. As a result, the non-recombining stretches of ‘supergene DNA’ evolve independently of one another, facilitating the rapid evolution of complex adaptations [36, 37]. These sets of genes are often polymorphic and subjected to balancing selection, as a species generally possesses at least two supergene variants [38]. Sex chromosomes, for instance, are classically considered supergenes [39]. In diploid organisms the heterogametic sex inherits the sex-determining ‘supergene’, as well as the alternate sex chromosome, in a hemizygous manner – meaning that they only receive one copy of each [40]. In the XX-XY sex determination system of mammals, for instance, males generally possess a single copy of the supergene that is the Y chromosome (as well as a single copy of the X-chromosome), whereas in the ZW-WW system it is the females that possess the Z supergene once (next to a single W chromosome). Hence, genes that are hemizygous and thus lie solely on the sex-determining supergene would show PAV in target capture datasets when data of different sexes is compared.

However, supergene systems are not limited to the biological concept of sex. Other, famous examples of supergenes underlying complex traits are; the Müllerian mimicry complex in *Numata* longwing butterflies [39], the striking sexual dimorphism and breeding behaviors of ruffs [41, 42] and white-throated sparrows [43], the social polymorphism observed in several species of ant [44], and heterostyly in primrose flowers [45]. Furthermore, hemizygous inheritance of (super)genes also occurs in, for instance, genetic incompatibilities such as with "hybrid necrosis" in plants, which can be linked to PAV in certain genes in for example Asian rice [46, 47]; hereditary diseases such as α-thalassaemia, which is caused by large deletions in the alpha globin gene cluster on chromosome 16 in humans [48]; and in balanced lethal systems, in which two distinct chromosome forms exist that are covered by unique lethal mutations [49].

We demonstrate the application of PAV-spotter using the balanced lethal system in *Triturus* newts as a case study. *Triturus* individuals either are heteromorphic and possess two different versions of their largest autosomal chromosome, characterized by unique deletions, or they are homomorphic and possess two identical versions of this chromosome [50]. The two types of homomorphic individuals express a unique disease state and both die during embryogenesis, whereas the heteromorphic individuals are viable [51-54]. This 'double hemizygous' system lends itself particularly well for using target capture data to detect PAV, as it allows for a reciprocal test: targets deleted from one chromosome version should be present on the alternate version and the other way around. Based on our findings, we describe the usefulness, as well as the limitations, of our approach.

Methods

Sample information & collection

The first chromosome of *Triturus* comes in two forms: 1A and 1B. Homomorphic individuals (1A1A or 1B1B) invariably die during embryogenesis, while heteromorphic individuals (1A1B/1B1A) are viable. We collected *T. macedonicus* x *T. ivanbureschi* F_1 hybrid embryos from an ongoing breeding experiment at the Institute for biological research, „Siniša Stanković", University of Belgrade [with experimental settings, breeding conditions, and other details on the process of raising embryos as described in; 55, 56]. Embryo development was followed through observation with a stereomicroscope. Diseased embryos were collected when the process leading up to developmental arrest occurred, which is visible as a 'growth slowdown', during the late tail-bud phase. Diseased embryos were then classified into either the "fat-tailed" (FT) phenotype or the "slim-tailed" (ST) phenotype based on morphological characteristics of the embryo [53, 57]. Healthy/control (HC) embryos that survived this critical phase were subsequently collected. We collected 30 individuals in total (Supplementary Table 1); ten of each class, i.e. ten ST, ten FT and ten HC embryos. Samples were stored in ethanol at -20 °C until further handling.

Laboratory procedures & pre-processing of sequence data

We followed the standard "NewtCap" workflow of salt-based extraction of DNA from embryonic tissue, followed by quantification, library preparation, target capture and Illumina sequencing [as described in; 58]. After obtaining the raw, paired-end sequence reads from Baseclear B.V. (Leiden, the Netherlands), we followed a standard pipeline for checking the quality of, and for cleaning-up and mapping, our sequence data in a Linux environment up to and until the deduplication of the BAM files step [as described in; 58]. These deduplicated BAM files served as input for further data extraction and analyses. Throughout the cleaning and mapping process, we used SAMtools' [59] *stats*, *flagstat* and *coverage* options to calculate basic statistics from the FASTQ and BAM files. The reference FASTA file used for read mapping can be found in Supplementary Material as 'Targets.fasta'. These 7,139 sequences, initially used for probe tiling, were based on *T. dobrogicus* transcripts, and had a maximum length of 450bp [58, 60].

Preparing read depth data for PAV-spotter

From the BAM files we extracted information on sequence read depth for all sites per target by using the SAMtools depth option [59]. We optimized this extracted information by following several file-manipulation steps in a custom 'prepping' shell script (Script 1) to make the input files and folder structure match the requirements of our PAV-spotter tool. The steps in Script 1 include automatically merging and sorting of intermediate files where appropriate, changing the tab-delimited format to a CSV format, splitting the overall CSV file into multiple files (one file per separate target/gene), and creating a text file with sample names for later use (details are explained in the script). The exact format of the input folder structure and input files is described in Box 1.

PAV-spotter assumes background knowledge on phenotype classes that presumably differ in the presence of certain genes (in other words, cross-comparisons are not random: for instance males are compared to females, diseased individuals are compared to control samples, etc.). Here, we work with the a priori classification of three phenotypes: two types of diseased embryos (FT vs. ST) and healthy embryos to serve as a control (HC). Which sample belongs to which class needs to be designated by your input filenames (the BAM files), which will end up in an automatically created text file "individuals.txt" after running Script 1. This is crucial, as PAV-spotter performs the comparisons based on the phenotype information embedded in the names of this text file. In case phenotypic classes as specified by the user cannot be deduced from input file names, the user needs to either alter the input file names manually, or alter the identifiers in Script 1 manually – or both – before running Script 1 (ideally, the user includes such filename identifiers at the raw FASTQ file stage for consistency throughout the pipeline).

Applying PAV-spotter

We applied Script 1 on our total set of 30 samples (Supplementary Table 2; n=30, ten ‘ST’, ten ‘FT’ and ten ‘HC’ individuals, with these identifier abbreviations occurring in the sample names). Additionally, we applied the script on random subsets of samples (Supplementary Table 3 and Supplementary Table 4); two separate analyses with a sample size of five per class (i.e. two total subsets of n=15, indicated by sample set ‘5_1’ and run ‘5_2’), and five more separate analyses with a sample size of two per class (i.e. five total subsets of n=6, indicated by sample set ‘2_1’, ‘2_2’, ‘2_3’, ‘2_4’ and ‘2_5’). This allowed us to assess the performance of PAV-spotter when lower sample sizes are used. We randomized the grouping of samples into subsets by using the ‘shuf’ command from the standard GNU Core Utilities (http://gnu.org/s/coreutils/).

We ran a custom MATLAB script (Script 2, hereafter ‘PAV-spotter’) remotely through SLURM workload manager (example batch script attached). Users can compare two, or three phenotypic classes (using the argument “categories”, see the SLURM script), but if only two are provided, only two are compared. Also, we indicated which class we considered the control group (by implementing “ctrl_category = 'HC'”) and we provided a common identifier for all input files/targets (with the argument “common_identifier = 'DN'”). Also, information on the working directory and desired output filename was provided in the MATLAB command depending on the analysis, and the same is the case for the customized #SBATCH lines for running the SLURM job.

BOX 1:
This is a description of the expected input file format for the PAV-spotter script:

‘Species’ Directories:
- Each species, or otherwise distinguishable set of data, has its own directory
- PAV-spotter is built in such a way that it will loop over multiple such directories
“individuals.txt” file:
- An automatically generated file, uses the initial sample names for input
- Needs to be located in, and corresponding to the contents of, a particular species directory
- This file contains the individual information, with each sample name on a new line and with information on the classes to be compared included in the name (e.g. ‘ST’, ‘FT’, and ‘HC’)
Gene/Target Data Files:
- Also needs to be located in, and corresponding to the contents of, a particular species directory
- Each file represents a single gene or target
- Each file has columns and is in CSV format (this should be the output from batch script 1):
- Column 1: Gene/target name
- Column 2: Gene/target position (a number)
- Column 3: Read depth data (a number)
- Column 4: Sample/class name (should match identifiers in “individuals.txt” file)

The script was run four times on our total set of n=30: once with default settings (no filtering, “reads_threshold” == 0 and “contig_width” == 0), one time with a mild coverage filtering (“reads_threshold” == 5 and “contig_width” == 0), one time with a mild filtering for minimum length of contigs (“reads_threshold” == 0, “contig_width” == 50) and one time with both of the filtering thresholds (“reads_threshold” == 5 and “contig_width” == 50). By setting a soft coverage filter of a minimum of five reads, we filter out reads of any poorly covered target of an individual that does not meet this criterium (thereby assuming a coverage of zero across the target in question). Furthermore, by specifying a minimum contig width, the script will filter out read information of covered regions in between two positions with zero coverage, in case those regions are narrower than, in this case, 50bp (thereby assuming a coverage of zero in that specific target region). For the tests on sample subsets of five individuals per class and two individuals per class, we only used the default filtering settings (no filtering, “reads_threshold” == 0 and “contig_width” == 0). In all analyses, we enabled the script to plot accessory figures (“plot_figures” == TRUE, clarification below). MATLAB v.9.13.0 (https://www.mathworks.com) was used for running PAV-spotter.

The rationale behind PAV-spotter

After setting up the input files and initiating PAV-spotter successfully, filtering settings are applied as specified. Subsequently, PAV-spotter merges and normalizes the sequence read depth information for all the samples per target per specified sampling class before it starts the actual cross-correlation analyses (but users can turn this setting off in case calculations of all possible cross-comparisons on an individual level are desired). By exploiting the availability of multiple samples per phenotypic class in this way, we ensure that the comparisons will be made on a class level rather than on an individual level. In the latter case, more false outcomes would be expected as a result of the stochastic nature of target capture experiments, something that can be avoided by pooling results. PAV-spotter can loop through a set of input directories if separate datasets need to be analyzed with similar settings consecutively

The cross-correlation analysis in PAV-spotter works as follows (Figure 1). Two target capture results of the same targeted region, but of a different phenotypic class, are defined by D_1 and D_2. These represent two vectors of identical length T, where the vector position represents the target position and the vector values represent the normalized sequence read depth. PAV-spotter then calculates the similarity of the cross-correlation formulated by

$$C_{1,2}[n] = \sum_{m=-T}^{T} D_1[m]D_2[m+n]$$

for all n between $-T$ and T, with n and m accessing the indices of the vectors $C_{1,2}, D_1, D_2$. This constructs a cross-correlation vector of which the maximum value provides the maximum similarity of D_1 and D_2. For example, in the case that $D_1 = D_2$ (i.e., in the case of autocorrelation), the similarity score C will be 100%.

PAV-spotter outputs a CSV file per separate analysis with all the cross-correlation data in the form of percentage similarity, and it outputs a folder with figures. Also, a CSV file with cross-comparisons between data from samples from the control group only (the HCs) is generated. This extra file allows for a calculation of the overall resemblance of the control samples, which should have data present with well-captured targets. This as opposed to the FT and ST samples, which are expected to show 'data gaps', or absence, for some targets.

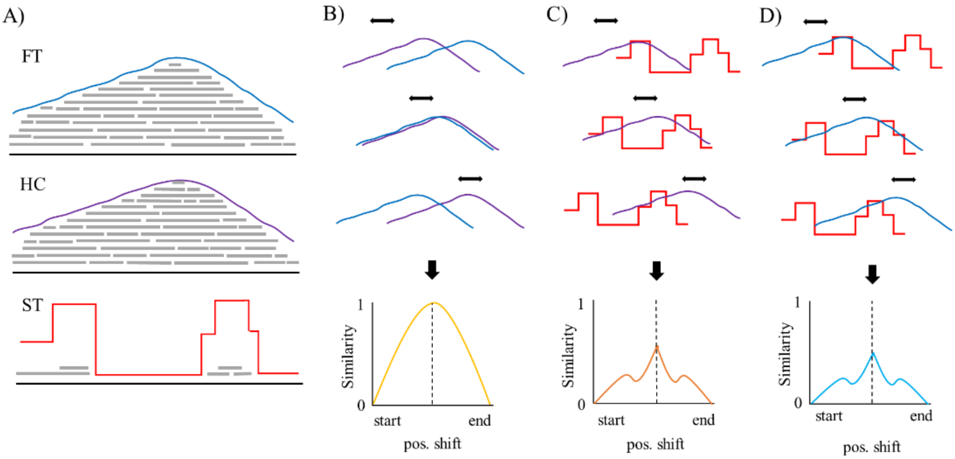


Figure 1:** A simplified visualization of the cross-correlation methodology in PAV-spotter, showing a hypothetical gene/target that is missing in only ST embryos as an example.* ***(A) *A number of sequence reads are mapped (in grey) against a gene/target. This information is taken from depth files and merged per phenotypic class (in case of multiple samples per class), then the absolute distributions of the read depths are normalized, here represented by the colored lines; dark blue = FT, purple = HC, red = ST.* ***(B)*** *The distribution data of the FT and the HC classes are compared. A measure of similarity is determined as a function of the displacement of one read depth distribution relative to the other, as if they were to 'slide over' each other (indicated by black arrows). The similarity appears close to 1 (=100%) and the graph produced by PAV-spotter also follows a smooth line.* ***(C)*** *The distribution data of the HC and ST classes are compared: the similarity is not close to 100% and the similarity graph produced by PAV-spotter does not follow a smooth line.* ***(D)*** *The distribution data of the HC and ST classes are compared: the similarity is not close to 100% and the similarity graph produced by PAV-spotter does not follow a smooth line.*

As we are investigating a double hemizygous system, we always have three cross-correlation values to work with. This means we are able to use not only the healthy embryos (HC), but also the other class of diseased embryo (ST or FT), as a control, because genes absent in one class of diseased embryo are expected to be present in both other embryo classes (e.g. to recognize absence in ST embryos, which should have a 1A1A genotype, we can check for presence in the HC embryos which should have the 1A1B genotype, but we can do an additional check for presence in the ST embryos that should have a 1B1B genotype – and the same applies the other way around). Hence, to deduce PAV in the chromosome that is inherited twice in ST embryos, we search for a pattern in which a significant portion of the target was present in both HC and FT embryos (which contain the alternate chromosome form), but absent in ST embryos. Conversely, to deduce PAV in the chromosome inherited twice by FT embryos, we searched for absence in FT, but presence in both HC and ST embryos.

Downstream PAV estimation

To automatically deduce PAV patterns, we applied another custom shell script, Script 3. This script takes the main output file of PAV-spotter, creates an overall matrix of the results, adds columns with information on the cross-correlation data that stand out, and makes lists of the targets that show potential PAV based on a threshold (which can be customized). When, for a certain target, the data of FT versus ST embryos were less than 80% similar, the data of FT versus HC embryos were less than 80% similar, and the data of ST versus HC embryos were more than 80% similar, this target was scored as '1A-linked'. For '1B-linked' genes it was the other way around: FT vs. ST embryos < 80% similar, ST vs. HC embryos < 80% similar, and FT vs ST embryos > 80% similar. This threshold of 20% dissimilarity equals a p-value of between 0.01 (≈ 15% dissimilarity) and 0.001 (≈ 28% dissimilarity). The PAV-spotter output file that shows all similarity scores among the HC embryos only, guided our choice for this threshold (e.g. see Figures 1 and 2).

Finally, as not much is known about the genetic background of our non-model study species *Triturus*, we performed visual inspections of the read content of all BAM files (n=30) that were used as input for PAV-spotter by checking them in Integrative Genomics Viewer (IGV) software (Robinson et al, 2011) on a Windows environment. This constitutes the last step in our overall workflow (Figure 2). We automated obtaining screenshots through IGV by running batch commands for all 7,139 targets (an example batch script is available). With a 'checking-by-eye' approach we categorized PAV hits as 'likely true' and 'likely false' in order to assess the quantity of false positive outcomes and we cross-checked the results of all the different runs to identify false negative outcomes (in other words: in case a 'likely true' target with PAV was retrieved in one analysis, but not in another, we counted it as a false negative in the latter analysis).

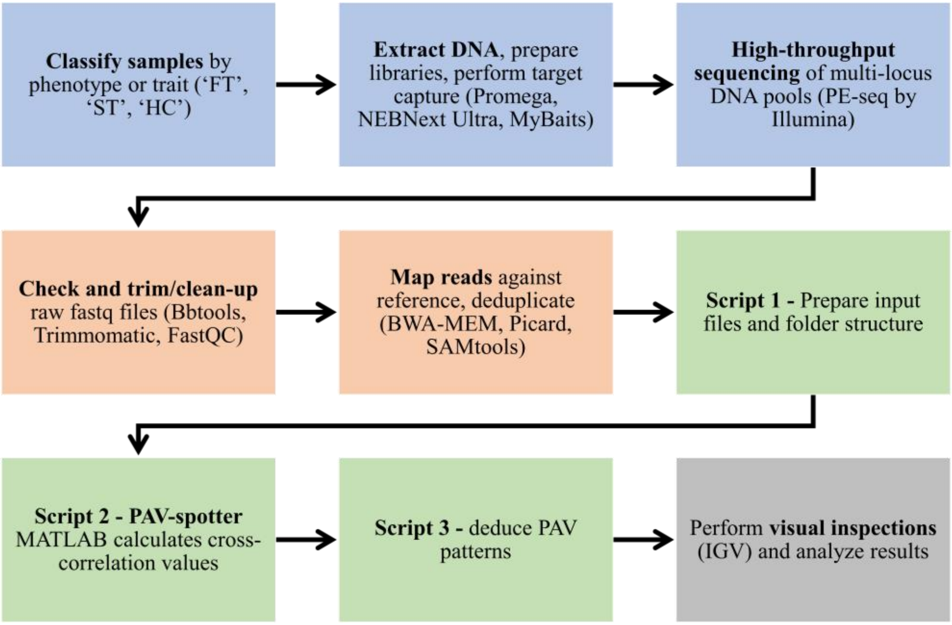

Classify samples by phenotype or trait ('FT', 'ST', 'HC')
Extract DNA, prepare libraries, perform target capture (Promega, NEBNext Ultra, MyBaits)
High-throughput sequencing of multi-locus DNA pools (PE-seq by Illumina)
Check and trim/clean-up raw fastq files (Bbtools, Trimmomatic, FastQC)
Map reads against reference, deduplicate (BWA-MEM, Picard, SAMtools)
Script 1 - Prepare input files and folder structure
Script 2 - PAV-spotter MATLAB calculates cross-correlation values
Script 3 - deduce PAV patterns
Perform **visual inspections** (IGV) and analyze results

Figure 2: *A summary of the consecutive steps of our methods in which the order of the steps is indicated by black arrows. Blue boxes show the laboratory process, orange boxes represent bioinformatic pre-processing steps, green boxes stand for the application of the main PAV-spotter scripts, and the grey box covers the conclusive steps of inspecting and interpreting the results.*

We executed all bioinformatic steps, from pre-processing of reads and read depth information to applying PAV-spotter and extracting information from the output, through the High Performance Computing facility called 'ALICE' (Academic Leiden Interdisciplinary Cluster Environment, the Netherlands). The GitHub repository of PAV-spotter provides all the scripts and further explanation: https://github.com/Wielstra-Lab/PAVspotter.

Results

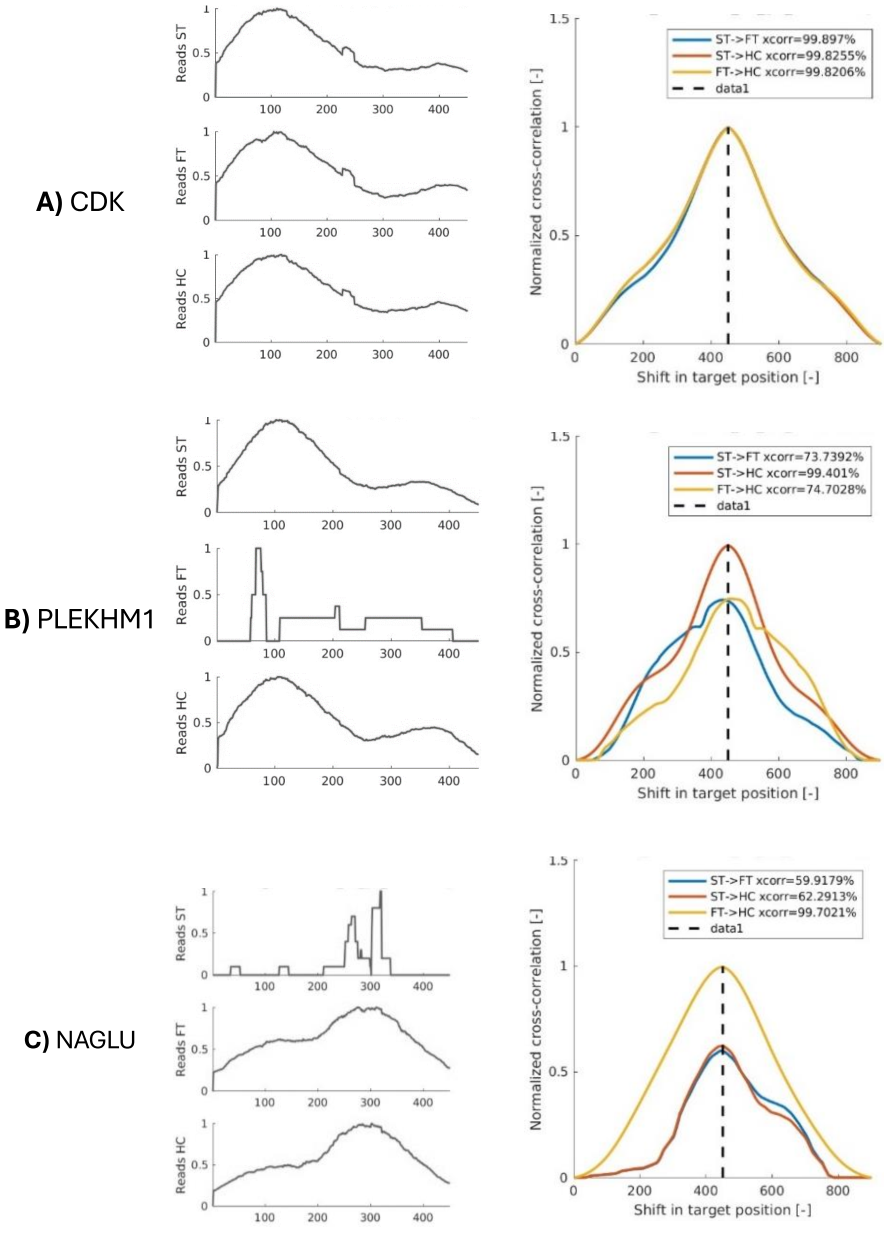
A mean of 6,483,062 read pairs were generated on average per sample, with a standard deviation of 1,450,648 read pairs (Supplementary Table 1). After trimming, this changed to a total of 6,187,680 read pairs with a SD of 1,369,254 read pairs. On average, 35.57% of the trimmed reads were successfully mapped against the reference targets after duplicate removal, as an average of 17.09% of all trimmed reads were flagged as duplicates (Supplementary Table 1).

Overall, the targets had a mean read depth of 90.09 sequences and a mean coverage of 97.19 % of the sequence bases (Supplementary Table 2, presented per phenotypic class). For the overall set with ten samples per phenotypic class, the average depth of coverage was 84.7 in the FT group, 97.7 in the group of HC embryos, and 87.8 in the ST embryo group (Supplementary Table 2). Moreover, for the batched samples with five individuals per phenotypic class, this average depth of coverage varied between the lowest number of 76.6 in FT batch 5-2 and the highest number of 102.2 in HC batch 5-1 (Supplementary Table 3). Lastly, for the batched samples with two individuals per phenotypic class, the averages varied between the lowest number of 37.1 in FT batch 2-1, and the highest number of 134.1 in FT batch 2-5 (Supplementary Table 4).

Through the different runs with a sample size of ten per class, we discovered large-scale PAV for in total 72 targets. Genes without PAV had sequence reads with similar distributions for all three classes (Figure 3A), whereas of those 72 aberrant genes that we discovered, 32 showed an absence in FT, but a presence in HC and ST embryos (and are thus "1A-linked", Table 5 and Figure 3B), while the remaining 40 were absent in ST, but not in HC and FT embryos (and are thus "1B-linked", Table 6 and Figure 3C). We confirmed our findings using the visual IGV inspection (for examples, see Supplementary Figures 3-5).

After running PAV-spotter without any filtering options on the full dataset, we correctly identified all 32 1A targets and generated one false positive in the 1A list, which we identified by the visual IGV inspection (see Supplementary Figure 6). By re-running PAV-spotter with the previously described filtering options, this false positive was removed from the list in some cases, however these additional analyses also generated more false positives and false negatives: the frequency of them depending on the combination of filtering settings used (Supplementary Table 5). For the 40 1B targets, we discovered 37 true positives and three false positives in the unfiltered run. Depending on the combination of filtering settings applied, two of these three false positives again disappeared from the list. However, these extra runs with filtering settings also highlighted three additional 1B targets that were overlooked (as false negatives) in the initial analysis (Supplementary Table 6).

The false positive outcomes consistently had either the lowest - or in a single case, the highest - mean depth values, reflected by the 'MAXpeak' output of the overall results matrix generated by PAV-spotter. This value is the peak number of reads in one position of a certain target observed across all the individual samples included in the analyses. The false negative outcomes that came to light as true positive results after additional filtering was applied, all had similarity scores for the resembling classes (FT and HC) above 99% in the unfiltered analysis. But the lower similarity scores of these targets between the non-resembling classes (both between FT and ST and between ST and HC) were lower than 90%, but not lower than the 80% threshold (which is why they were initially overlooked).


A) CDK
Reads ST
Reads FT
Reads HC
ST->FT xcorr=99.897%
ST->HC xcorr=99.8255%
FT->HC xcorr=99.8206%
data1
Normalized cross-correlation [-]
Shift in target position [-]
B) PLEKHM1
Reads ST
Reads FT
Reads HC
ST->FT xcorr=73.7392%
ST->HC xcorr=99.401%
FT->HC xcorr=74.7028%
data1
Normalized cross-correlation [-]
Shift in target position [-]
C) NAGLU
Reads ST
Reads FT
Reads HC
ST->FT xcorr=59.9179%
ST->HC xcorr=62.2913%
FT->HC xcorr=99.7021%
data1
Normalized cross-correlation [-]
Shift in target position [-]

◀ ***Figure 3:*** *Examples of target-specific plots produced by PAV-spotter in the overall run with no filtering (n=10 per class). The stacked plots on the left side within the panels A-C show the merged and normalized distributions of sequence read depths per phenotypic class, and the colored plots on the right side within the panels A-C show associated measures of similarity of those distributions as a function of the displacement of one relative to the other (including a legenda explaining the colors and cross-correlation values).* ***A)*** *An example of a 'normal' gene/target, showing the cross-correlation analyses of control marker 'CDK' (see Discussion), with a similar shape of the sequence read depth distributions and high correlation values (above the 80% similarity threshold) between all three classes.* ***B)*** *An example of a 1A-linked gene/target, showing the cross-correlation analyses of 'PLEKHM1' (see Discussion), with a deviant read depth distribution for the FT class and a high correlation value (>80%) for ST vs. HC samples, but a lower cross-correlation value (<80%) for the ST vs. FT and FT vs. HC sample comparisons.* ***C)*** *An example of a 1B-linked gene/target, showing the cross-correlation analyses of 'NAGLU' (see Discussion), with a deviant read depth distribution for the ST class and a high correlation value (>80%) for FT vs. HC samples, but lower cross-correlation values (<80%) for the FT vs. ST and the ST vs. HC samples.*

The results for the n=5 per class runs (5_1 and 5_2) resemble our earlier findings. We re-discovered an average of 68.5 (95.1% success) out of the total of 72 PAV targets previously discovered, but with half the sample size. For run 5_1, this number was 70 out of 72 PAV targets (97.2% success) and for run 5_2 this number was 67 out of 72 (93.1% success). Overall, for 1A, the results of both the analyses with a sample size of five individuals per class were complementary, as all 32 previously identified 1A targets were re-discovered at least once (Supplementary Table 7). The same goes for the 40 previously identified 1B targets (Supplementary Table 8). Between the two runs, the overall mean depth of coverage was the lowest in 5_2, the analyses that also showed less successful out of the two.

The analysis of the 5_1 subset (n=15) resulted in the discovery of 31 true 1A targets plus one false negative and one two positives, and in 39 true 1B targets with one false negatives and one false positive. The analysis of the 5_2 subset (n=15) again resulted in 31 true 1A targets (with one difference) plus one false negative and two false positives. These false outcomes were not the same as with the 5_1 subset analysis. For 1B, the 5_2 analysis yielded 36 true 1B targets with four false negatives and seven false positives.

The results of the n=2 per class runs (2_1 through 2_5, each with n=6 in total) again highlighted the same PAV exhibiting targets. We re-discovered an average of 70 (97.2% success) out of the total of 72 PAV targets previously discovered, but with a fifth of the sample size (Supplementary Tables 9 and 10). For each of the 2_1, 2_2 and 2_4 runs, these numbers were indeed 70 out of 72 (97.2% success), run 2_3 retrieved 71 out of 72 PAV targets (98.6% success) and run 2_5 69 out of 72 (95.8%).

Overall, for both the 32 previously identified 1A targets and the 40 previously identified 1B targets, these five analyses with a sample size of only two individuals per class appeared complementary, as all true positive targets were re-discovered at least once. For 1A, one of the false negative outcomes came forward as a false negative in two out of the five analyses (and as a true positive in the three other analyses). The other three false negative outcomes in the 1A list were incidental. For 1B, each of the false negative outcomes occurred in only one of the five analyses (and formed a true positive result in the four alternative analyses). The number of false positive outcomes was slightly higher with these low sample size tests (Supplementary Tables 11 and 12), however this was especially noticeable for the 1A results of the first batch (2_1). The mean depth of coverage was also the lowest for the FT (1B1B) samples in this batch (Supplementary Table 4). We therefore tested for a correlation and show that the mean depth of the samples exhibiting absence (i.e. the mean depth of FT samples with determining 1A absence, and the mean depth of the ST with determining 1B absence) appeared to be negatively and significantly correlated to the number of false positives brought forward (Spearmann's rank correlation, n=10, p=0.018).

Discussion

We employ a signal cross-correlation approach to discern PAV patterns in target capture data of *Triturus* newt DNA. By comparing the read depth in sequence data of embryos of different phenotypic classes, and by manually checking the results of the read alignments, we are able to identify over seventy targets that appear to be either present in, or absent from, the genome, depending on the phenotype.

The three example targets displayed in Figure 3 have been independently tested using multiplex (mx) PCR techniques, including mxKASP [61]. Control marker CDK is present in all three embryo classes. This corresponds with our results from PAV-spotter, where we discover high cross-correlations values between the distribution of mapped reads of all three phenotypic classes for this target marker (>99% similarity, way above our 80% cutoff threshold). On the other hand, PLEKHM1 is observed in HC and ST embryos, but not in FT embryos, and NAGLU is observed in HC and FT embryos, but not in ST embryos [61]. Again, this matches our PAV-spotter findings – even with sample sizes as low as two individuals per phenotypic class.

Evidently, PAV-spotter relies on the correct, a priori classification of samples. Also, it merges the read depth data of samples per each phenotypic class in case multiple samples are provided (a setting that is recommended, but can be turned off if desired). Thus, future users should carefully consider what they want to compare. Additionally, we underline that cross-contamination of DNA is, for example, a main concern with target capture of ancient DNA [62] and it could potentially distort the similarity values calculated by PAV-spotter – something to be wary of.

The fact that PAV-spotter is able to, on average, re-discover 97,2% of true positive PAV target outcomes in our trials is especially convenient for studies where scientists must rely on a limited amount of available DNA, as is often the case with herbarium specimens [63, 64]. However, the highest yield of false positive outcomes is observed in our 1A results of batch 2_1, but not the accessory 1B results, which firstly shows that there is a likely trade-off between sample size and sequence coverage. Preferably, the quality of DNA is as high as possible when working with small sample sizes. Although also preferred in case of a larger sample size per phenotypic class, the chances are then higher that any poor coverage sample(s) will be compensated for by sample(s) with better coverage. In general, spreading sequencing efforts across at least a couple samples, with slightly lower – but still informative – depths per sample, is considered a safer option than working with extremely low sample sizes [65, 66].

Regardless of sample size, identifying and characterizing structural variation from target capture data is widely recognized to be difficult [67] and it is especially challenging with non-model organisms. This is because capture-rates may vary considerably depending on bait design, sample quality, species relatedness, batch effects, and stochastic factors [10, 68]. In most cases where targets showed a significant absence of mapped reads, we observe a small amount of reads being (mis)mapped against reference targets when none are expected, for instance (visible in the PAV-spotter output figures and the IGV screenshots). Occasionally, these consist of (clipped/partially matching) reads, something that can be caused by sequencing errors, chimeric reads, errors in the reference sequence, tandem duplications, or genomic rearrangements and structural variants [69, 70].

A solution to remove any unwanted (mis-)mapped reads would be to filter more strictly upstream. However, in case of samples or targets that show poor or limited coverage – an issue that is not uncommon with target capture procedures [71] – strict filtering may not be desired. This means that, due to this potential stochasticity, merely using existing tools to check whether there are any mapped reads at all in a sample/target (i.e. checking for the presence of zero reads versus >0 reads), or building *de novo* assembled contigs per target on a sample-per-sample basis, will not be sufficient to identify PAV accurately. Our method offers an alternative solution, as PAV-spotter appears robust enough to detect PAV, even in the face of low coverage and (partially) mis-mapped and clipped reads. However, the trade-off in false positive and false negative outcomes will largely depend on the similarity thresholds and other criteria set by users, as well as on any manual, double-checks performed.

In conclusion, we show that considering the genomic position as a variable for signal displacement instead of time – which is generally the case in more classic cross-correlation applications [29] – makes it possible to identify markers of PAV/structural variation in target capture data, without needing any prior knowledge on large-scale, genomic context. Our study shows that a multidisciplinary bioengineering and biotechnological approach can help bioinformatics, and thus the fields of evolutionary and molecular research, forward when dealing with challenging research questions, datasets, and study organisms.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 802759). Sample collection was supported by the Serbian Ministry of Science, Technological Development and Innovation (grants nos. 451-03-66/2024-01/200007, 451-03-65/2024-03/200178, 451-03-66/2024-03/200178). Dr. Ana Ivanović supported breeding and collecting *Triturus* samples. Individuals of *T. ivanbureschi* and *T. macedonicus* were originally collected from natural populations: *T. ivanbureschi* from Zli Dol, Serbia with permission obtained from the Serbian Ministry of Energy, Development and Environmental Protection (permit No. 353-01-75/2014-08) and *T. macedonicus* in Ceklin, Montenegro with permission obtained from the Agency for Environmental Protection, Montenegro (permit No. UPI-328/4).The experimental procedures were approved by the Ethics Committee of the Institute for Biological Research „Siniša Stanković", University of Belgrade (decisions no. 03-03/16 and 01-1949). All experiments were performed in accordance with the Directive 2010/63/EU.

Data accessibility and Benefit-sharing statement

The raw, Illumina sequencing reads used in this study have been submitted to the NCBI Sequence Read Archive (SRA) and are publicly available through BioProject number 'PRJNA1111729' (https://www.ncbi.nlm.nih.gov/sra/PRJNA1111729). The PAV-spotter tool, including further explanation on how to use and customize the code, is available through the GitHub repository: https://github.com/Wielstra-Lab/PAVspotter. Also, certain Supplementary Materials are provided through an online Zenodo repository: https://zenodo.org/records/13991751. Figures created by PAV-spotter for each separate analysis that are not shown in the paper can be provided upon request. The same goes for all IGV screenshot images generated by the batch script.

Author contributions

MdV and CvdP designed the tool. BW, MdV and CvdP designed the experiments. MC, TV and AI collected the samples and conducted the phenotypic classification. MdV and AT performed the molecular laboratory work. CvdP wrote the PAV-spotter script (Script 2), which comprises of the signal cross-correlation code. Other scripts (Script 1, Script 3, and the IGV batch script) were written by MdV. MdV conducted the main bioinformatics, data acquisition, and data interpretation. MdV drafted the work and CvdP added mathematical details to the text and conducted p-value/threshold estimations. All authors revised the manuscript and approved of the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Annavi, G., J.E. Uli, and R. Arumugam, *A review of the application of Next Generation Sequencing (NGS) in wild terrestrial vertebrate research.* Annual Research & Review in Biology, 2019: p. 1-9.
2. Morganti, S., et al., *Next generation sequencing (NGS): a revolutionary technology in pharmacogenomics and personalized medicine in cancer*, in *Translational Research and Onco-Omics Applications in the Era of Cancer Personal Genomics*, E.A.-d.l.V. Ruiz-Garcia, H., Editor. 2019, Springer Nature Switzerland: Advances in Experimental Medicine and Biology. p. 9-30.
3. Lemmon, E.M. and A.R. Lemmon, *High-throughput genomic data in systematics and phylogenetics.* Annual Review of Ecology, Evolution, and Systematics, 2013. **44**(1): p. 99-121.
4. Ekblom, R. and J. Galindo, *Applications of next generation sequencing in molecular ecology of non-model organisms.* Heredity, 2011. **107**(1): p. 1-15.
5. Wachi, N., K.W. Matsubayashi, and K. Maeto, *Application of next-generation sequencing to the study of non-model insects.* Entomological Science, 2018. **21**(1): p. 3-11.
6. Zaharias, P., et al., *Data, time and money: evaluating the best compromise for inferring molecular phylogenies of non-model animal taxa.* Molecular Phylogenetics and Evolution, 2020. **142**: p. 106660.
7. Rovelli, V., et al., *Genotyping-by-Sequencing (GBS) of large amphibian genomes: a comparative study of two non-model species endemic to Italy.* Animal Biology, 2019. **69**(3): p. 307-326.
8. Etherington, G.J., et al., *Sequencing smart: de novo sequencing and assembly approaches for a non-model mammal.* Gigascience, 2020. **9**(5).
9. Da Fonseca, R.R., et al., *Next-generation biology: sequencing and data analysis approaches for non-model organisms.* Marine Genomics, 2016. **30**: p. 3-13.
10. Andermann, T., et al., *A guide to carrying out a phylogenomic target sequence capture project.* Frontiers in Genetics, 2019. **10**: p. 1407.
11. Puritz, J.B. and K.E. Lotterhos, *Expressed exome capture sequencing: a method for cost-effective exome sequencing for all organisms.* Molecular Ecology Resources, 2018. **18**: p. 1209-1222.
12. Kaur, P. and K. Gaikwad, *From genomes to gene-omes: exome sequencing concept and applications in crop improvement.* Frontiers in Plant Science, 2017. **8**: p. 1-7.
13. Yohe, L.R., et al., *Evaluating the performance of targeted sequence capture, RNA-Seq, and degenerate-primer PCR cloning for sequencing the largest mammalian multigene family.* Molecular Ecology Resources, 2020. **20**: p. 140-153.
14. Bi, K., et al., *Transcriptome-based exon capture enables highly cost-effective comparative genomic data colection at moderate evolutionary scales.* BMC Genomics, 2012. **13**(403): p. 1471-2164.
15. Gauthier, J., et al., *A brief history of bioinformatics.* Briefings in Bioinformatics, 2019. **20**(6): p. 1981-1996.
16. Johnson, M.G., et al., *HybPiper: Extracting coding sequence and introns for phylogenetics from high-throughput sequencing reads using target enrichment.* Applications in Plant Sciences, 2016. **4**(7).
17. Yuan, H., et al., *Assexon: assembling exon using gene capture data.* Evolutionary Bioinformatics, 2019. **15**: p. 1176934319874792.
18. Andermann, T., et al., *SECAPR-a bioinformatics pipeline for the rapid and user-friendly processing of targeted enriched Illumina sequences, from raw reads to alignments.* PeerJ, 2018. **6**: p. e5175.
19. Zhang, L.M., et al., *Genome-wide patterns of large-size presence/absence variants in sorghum.* Journal of Integrative Plant Biology, 2014. **56**(1): p. 24-37.
20. Wang, Y., et al., *Exploration of presence/absence variation and corresponding polymorphic markers in soybean genome.* Journal of Integrative Plant Biology, 2014. **56**(10): p. 1009-19.
21. Wold, J., et al., *Expanding the conservation genomics toolbox: incorporating structural variants to enhance genomic studies for species of conservation concern.* Molecular Ecology, 2021. **30**(23): p. 5949-5965.
22. Mahmoud, M., et al., *Structural variant calling: the long and the short of it.* Genome Biology, 2019. **20**(1): p. 246.
23. Marroni, F., S. Pinosio, and M. Morgante, *Structural variation and genome complexity: is dispensable really dispensable?* Current Opinion in Plant Biology, 2014. **18**: p. 31-6.
24. Gerdol, M., et al., *Massive gene presence-absence variation shapes an open pan-genome in the Mediterranean mussel.* Genome Biology, 2020. **21**(1): p. 275.

25. Hu, H., et al., *Amborella gene presence/absence variation is associated with abiotic stress responses that may contribute to environmental adaptation.* New Phytologist, 2022. **233**(4): p. 1548-1555.
26. Rosa, R.D., et al., *High polymorphism in big defensin gene expression reveals presence-absence gene variability (PAV) in the oyster Crassostrea gigas.* Developmental and Comparative Immunology, 2015. **49**(2): p. 231-8.
27. Gabur, I., et al., *Gene presence-absence variation associates with quantitative Verticillium longisporum disease resistance in Brassica napus.* Scientific Reports, 2020. **10**(1): p. 4131.
28. Wellenreuther, M., et al., *Going beyond SNPs: the role of structural genomic variants in adaptive evolution and species diversification.* Molecular Ecology, 2019. **28**(6): p. 1203-1209.
29. Verhaegen, M. and V. Verdult, *4.3 Random Signals*, in *Filtering and system identification: a least squares approach.* . 2007, Cambridge university press. p. 100-103.
30. Gao, Z., C. Cecati, and S.X. Ding, *A survey of fault diagnosis and fault-tolerant techniques - Part I: fault diagnosis with model-based and signal-based approaches.* IEEE Transactions on Industrial Electronics, 2015. **62**(6): p. 3757-3767.
31. Jain, A.K., R.P.W. Duin, and J. Mao, *Statistical pattern recognition: A review.* IEEE Transactions on pattern analysis and machine intelligence, 2000. **22**(1): p. 4-37.
32. Hall, D.W. and M.L. Wayne, *Ohno's "peril of hemizygosity" revisited: gene loss, dosage compensation, and mutation.* Genome Biology and Evolution, 2013. **5**(1): p. 1-15.
33. Thompson, M.J. and C.D. Jiggins, *Supergenes and their role in evolution.* Heredity, 2014. **113**: p. 1-8.
34. Gutierrez-Valencia, J., et al., *The Genomic Architecture and Evolutionary Fates of Supergenes.* Genome Biology and Evolution, 2021. **13**(5).
35. Berdan, E.L., et al., *Genomic architecture of supergenes: connecting form and function.* Philosophical Transactions of the Royal Society B, 2022. **377**(1856): p. 20210192.
36. Pennisi, E., *'Supergenes' drive evolution.* Science, 2017. **357**: p. 1083.
37. Schwander, T., R. Libbrecht, and L. Keller, *Supergenes and complex phenotypes.* Current Biology, 2014. **24**: p. R288-R294.
38. Llaurens, V., A. Whibley, and M. Joron, *Genetic architecture and balancing selection: the life and death of differentiated variants.* Molecular Ecology, 2017. **26**: p. 2430-2448.
39. Joron, M., et al., *A conserved supergene locus controls colour pattern diversity in Heliconius butterflies.* PLoS Biology, 2006. **4**(10): p. e303.
40. Dufresnes, C. and P.A. Crochet, *Sex chromosomes as supergenes of speciation: why amphibians defy the rules?* Philosophical Transactions of the Royal Society B, 2022. **377**(1856): p. 20210202.
41. Lamichhaney, S., et al., *Structural genomic changes underlie alternative reproductive strategies in the ruff (Philomachus pugnax).* Nature Genetics, 2016. **48**(1): p. 84-8.
42. Kupper, C., et al., *A supergene determines highly divergent male reproductive morphs in the ruff.* Nature Genetics, 2016. **48**(1): p. 79-83.
43. Tuttle, E.M., et al., *Divergence and functional degradation of a sex chromosome-like supergene.* Current Biology, 2016. **26**: p. 344-350.
44. Kay, T., Q. Helleu, and L. Keller, *Iterative evolution of supergene-based social polymorphism in ants.* Philosophical Transactions of the Royal Society B, 2022. **377**(1856): p. 20210196.
45. Li, J., et al., *Genetic architecture and evolution of the S locus supergene in Primula vulgaris.* Nature Plants, 2016. **2**(12): p. 16188.
46. Li, J. and C.R. Lee, *The role of gene presence-absence variations on genetic incompatibility in Asian rice.* New Phytologist, 2023. **239**(2): p. 778-791.
47. Li, C., et al., *Tight genetic linkage of genes causing hybrid necrosis and pollinator isolation between young species.* Nature Plants, 2023. **9**(3): p. 420-432.
48. Harteveld, C.L. and D.R. Higgs, *α-thalassaemia.* Orphanet Journal of Rare Diseases, 2010. **5**(13).
49. Wielstra, B., *Balanced lethal systems.* Current Biology, 2020. **30**: p. R742-R743.
50. De Visser, M.C., et al., *Conserved gene content and unique phylogenetic history characterize the 'bloopergene' underlying Triturus' balanced lethal system* bioRxiv, 2024: p. 2024.10.25.620277.
51. Macgregor, H.C. and H. Horner, *Heteromorphism for chromosome 1, a requirement for normal development in crested newts.* Chromosoma, 1980. **76**: p. 111-122.
52. France, J., et al., *Genomic evidence suggests the balanced lethal system in Triturus newts originated in an instantaneous speciation event.* bioRxiv, 2024: p. 2024.10.29.620207.

53. Sessions, S.K., et al., *Cytology, embryology, and evolution of the developmental arrest syndrome in newts of the genus Triturus (Caudata: Salamandridae)*. The Journal of Experimental Zoology, 1988. **248**: p. 321-334.

54. Wallace, H., *Abortive development in the crested newt Triturus cristatus*. Development, 1987. **100**: p. 65-72.

55. Vučić, T., et al., *The reproductive success of Triturus ivanbureschi x T. macedonicus F_1 hybrid females (Amphibia: Salamandridae)*. Animals, 2022. **12**(4).

56. Vucic, T., et al., *A staging table of Balkan crested newt embryonic development to serve as a baseline in evolutionary developmental studies*. Journal of Experimental Zoology Part B: Molecular and Developmental Evolution, 2024.

57. Sims, S.H., et al., *Chromosome 1 in crested and marbled newts (Triturus) - An extraordinary case of heteromorphism and independent chromosome evolution*. Chromosoma, 1984. **89**: p. 169-185.

58. De Visser, M.C., et al., *NewtCap: an efficient target capture approach to boost genomic studies in Salamandridae (True Salamanders and Newts)*. bioRxiv, 2024: p. 2024.10.25.620290.

59. Danecek, P., et al., *Twelve years of SAMtools and BCFtools*. Gigascience, 2021. **10**(2).

60. Wielstra, B., et al., *Phylogenomics of the adaptive radiation of Triturus newts supports gradual ecological niche expansion towards an incrementally aquatic lifestyle*. Molecular Phylogenetics and Evolution, 2019. **133**: p. 120-127.

61. Meilink, W.R.M., et al., *Determining zygosity with multiplex Kompetitive Allele-Specific PCR (mxKASP) genotyping*. bioRxiv, 2024: p. 2024.10.25.620256.

62. Zavala, E.I., et al., *Quantifying and reducing cross-contamination in single- and multiplex hybridization capture of ancient DNA*. Molecular Ecology Resources, 2022. **22**(6): p. 2196-2207.

63. Kates, H.R., et al., *The effects of herbarium specimen characteristics on short-read NGS sequencing success in nearly 8000 specimens: old, degraded samples have lower DNA yields but consistent sequencing success*. Frontiers in Plant Science, 2021. **12**: p. 669064.

64. Hart, M.L., et al., *Retrieval of hundreds of nuclear loci from herbarium specimens*. Taxon, 2016. **65**(5): p. 1081-1092.

65. Pezzini, F.F., et al., *Target capture and genome skimming for plant diversity studies*. Applications in Plant Sciences, 2023. **11**(4): p. e11537.

66. Lou, R.N., et al., *A beginner's guide to low-coverage whole genome sequencing for population genomics*. Molecular Ecology, 2021. **30**(23): p. 5966-5993.

67. Jones, M.R. and J.M. Good, *Targeted capture in evolutionary and ecological genomics*. Molecular Ecology, 2016. **25**(1): p. 185-202.

68. Feng, Y., et al., *Improved molecular diagnosis by the detection of exonic deletions with target gene capture and deep sequencing*. Genetics in Medicine, 2015. **17**(2): p. 99-107.

69. Suzuki, S., et al., *ClipCrop: a tool for detecting structural variations with single-base resolution using soft-clipping information*. BMC Bioinformatics, 2011. **12**: p. S7.

70. Schroder, J., et al., *Socrates: identification of genomic rearrangements in tumour genomes by re-aligning soft clipped reads*. Bioinformatics, 2014. **30**(8): p. 1064-1072.

71. Bragg, J.G., et al., *Exon capture phylogenomics: efficacy across scales of divergence*. Molecular Ecology Resources, 2016. **16**(5): p. 1059-68.