Genome evolution of a pandemic lineage of the wheat blast pathogen

Angus George Gunnar Malmgren

Thesis submitted for the degree of Master of Science by Research

University of East Anglia

The Sainsbury Laboratory

Submitted 11th July 2025

Word count: 27,605

Abstract

Blast fungus, *Magnaporthe oryzae*, is a plant pathogen composed of multiple host-specific lineages and causes significant damage to cereal crops. The *Triticum* lineage is responsible for the highly damaging wheat blast disease prevalent within South America, which was identified in 1985. Whilst originally confined to South America, since 2016, a clonal lineage infecting on wheat has been found to have also spread to Zambia and Bangladesh, and was termed the B71 lineage after the South American B71 isolate which forms part of the group. To aid understanding of the adaptive evolution in this lineage, genomic segments in Zambian and Bangladeshi isolates which were not conserved with the B71 isolate were identified, and subsequently used to identify population-specific effector candidates. Two of these candidate effectors were investigated further: Art1_WB_ZM identified in the Zambian population, and APiasL3 identified in the Bangladeshi population. *APiasL3* is notable for being present in the mini-chromosome in Bangladeshi isolates in a region which has also been integrated into the core genome in one isolate. Following attempted generation of effector deletion transformants, leaf drop infection assays using *Art1_WB_ZM*-deletion transformants did not demonstrate altered ability to infect on different hosts compared to the wild type. Structures for both candidates were predicted using AlphaFold, and patterns of conserved residues were identified in the APiasL3 candidate with the family of proteins it belongs to, in addition to indications of positive selection on certain APiasL3 residues. Improving understanding of the effector makeup of blast fungus lineages could aid response strategy development and inform management practices to limit further disease spread.

Statement

I declare that none of the work presented within this thesis has previously been submitted for a degree at this or any University, and all material not produced by myself has been appropriately acknowledged.

Contents

Tables

Figures

Preface

This thesis documents my work performed in the Kamoun lab at The Sainsbury Laboratory in Norwich, UK, between Autumn 2021 and Summer 2024. It also includes some of the work I performed, again within the Kamoun group, between Autumn 2019 and Autumn 2021, which was structured as a predoctoral internship. Appendix 3 contains a brief outline of work I contributed to in addition to this thesis.

Acknowledgements

I would especially like to thank the following people:
Sophien Kamoun for supervision, invaluable feedback and guidance; Nick Talbot for feedback and supervision; Thorsten Langner for supervision and guidance, data, and many contributions to the project; Cristina Barragan for supervision and guidance, data, and many contributions to the project; Joe Win for guidance, data, and work on the project; Adeline Harant for project work, experimental guidance and assistance; Hernán Burbano for feedback and guidance; Sergio Latorre for technical guidance and advice; Michelle Hulin for contribution to the project; Ram Krishna Shrestha for technical guidance and project contributions; Vincent Were for providing additional genome sequence data and technical guidance; Aleksandra Białas for contribution to the project.

I would also like to thank all past and present members of the Kamoun group, and the broader TSL community and the TSL support teams.

This work was supported by the UKRI Biotechnology and Biological Sciences Research Council Norwich Research Park Biosciences Doctoral Training Partnership [grant number BB/ /T008717/1].

Chapter 1: Introduction

General introduction to the blast fungus and its population structure

Human actions such as trade and alteration of the environment are aggravating the already significant dangers caused by fungal pathogens to food security, human and animal health, and the environment, by inadvertently aiding their spread (Fisher *et al.*, 2012). The results of these outbreaks have included colony collapse disorder affecting bees, Sea-fan aspergillosis affecting corals and sizeable amphibian population declines across multiple continents resulting from chytridiomycosis (Fisher *et al.*, 2012).

Magnaporthe oryzae is a fungal pathogen considered to be amongst the most important plant pathogens, and is composed of multiple host-specific lineages able to infect cereal crops and wild grasses (Dean *et al.*, 2012). The relation between these different lineages is shown in Figure 1, reproduced from Gladieux *et al.* (2018). Once blast fungus conidia arrive on the surface of its host, a structure called the appressorium forms, which enables the fungus to penetrate into the plant cell (Dean *et al.*, 2012). After this point, hyphal growth can infect through multiple cells, secreting effector proteins which aid the colonisation process (Wilson and Talbot, 2009).

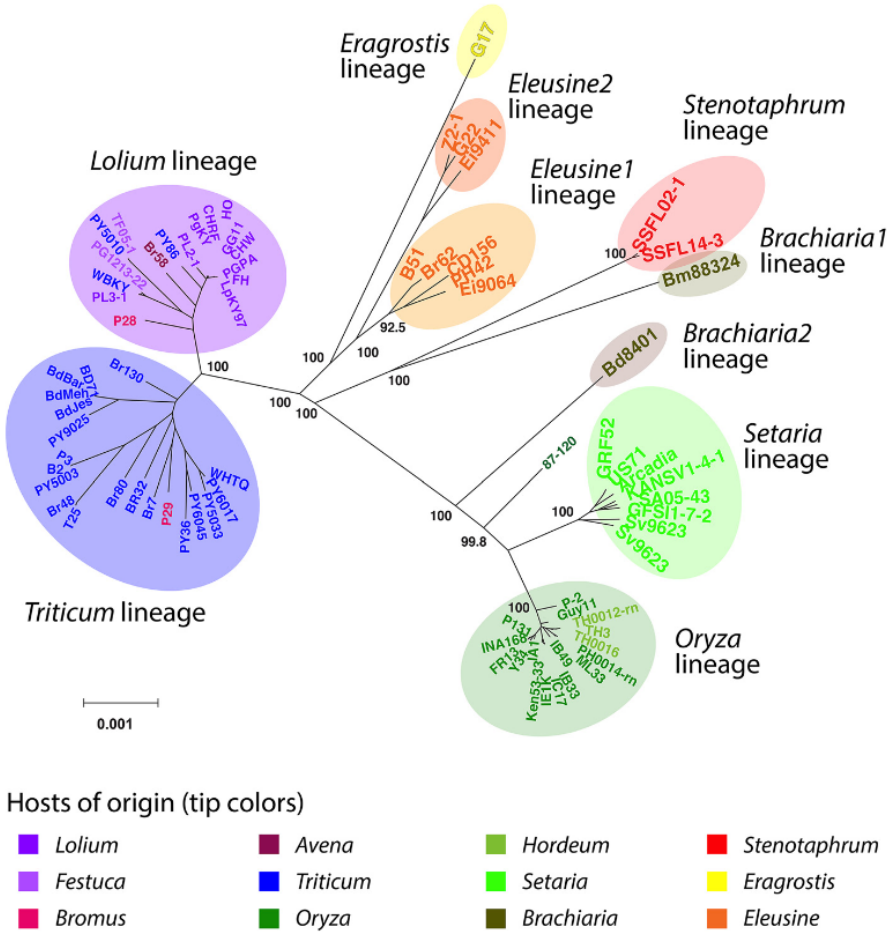

Figure 1. Magnaporthe oryzae host-specific lineages. Figure source: (Gladieux, Condon, et al.*, 2018)*

Perhaps the most famous example of blast fungus is the rice-infecting *Oryza* lineage, with yield losses typically between 10-30% (Talbot, 2003). Three clonal lineages of rice-infecting *M. oryzae* are dispersed around the globe, believed to have originated within the last 200 years from independent clonal expansions (Latorre *et al.*, 2020). As discussed by Latorre *et al.* each of these lineages has relatively low genetic diversity. A fourth population grouping, composed of genetically diverse isolates able to undergo recombination as part of sexual reproduction, has also been identified as being present primarily in southeast Asia (Latorre *et al.*, 2020), which is believed to be the centre of origin for the rice-infecting lineage (Saleh *et al.*, 2012). Presence of the clonal lineages around the world varies by region, with each of the lineages having a distinct fingerprint of effectors carried, and may be a critical component of adaptation to the host (Latorre *et al.*, 2020). As discussed by Latorre *et al.* the lineage with the largest effector complement is the diverse recombining population.

The emergence of wheat-infecting blast fungus

The wheat-infecting *Triticum* blast fungus lineage, which emerged in Brazil in 1985, is a significant threat to wheat production both in individual nations and to key wheat-producing regions (Cruz and Valent, 2017). After its emergence in Brazil, it has spread further in South America, including to Bolivia and Argentina (Cruz and Valent, 2017; Singh *et al.*, 2021). Wheat-infecting blast fungus has been shown to negatively impact grain quality and yield, and may inhibit formation of grain above the infection point (Cruz and Valent, 2017). Figure 2 illustrates typical bleaching symptoms of the wheat head (reproduced from Islam *et al.* (2020)).

Figure 2. Wheat Blast fungus bleaching symptoms on wheat heads. Figure source: (Islam et al.*, 2020).*

The emergence of wheat-infecting blast fungus in Bangladesh

In 2016, it was reported that wheat blast had appeared in Bangladesh (Malaker *et al.*, 2016). In the initial infection season, 15% of the land where wheat was grown was affected (Islam *et al.*, 2016). The initial infection season in 2016, in Jhenaidah district, had a 51% average yield loss (Islam *et al.*, 2016), although the yield loss significantly declined in subsequent infection seasons (Singh *et al.*, 2021). The organism responsible was subsequently found to be the same as the South American wheat blast fungus (Islam *et al.*, 2016).

The emergence of wheat-infecting blast fungus in Zambia

Wheat blast was first reported in 2020 to have emerged in Zambia in the 2017-2018 season, both in experimental plots and rain-fed fields (Tembo *et al.*, 2020). Later, the Zambian and Bangladeshi isolates were found to belong to the same lineage as each other (Latorre and Burbano, 2021), and they, together with the South American B71 isolate, form a pandemic clonal lineage, which together is termed the "B71 lineage" (Latorre *et al.*, 2023).

The successive movements of wheat blast fungus are illustrated in Figure 3, from its initial spread in South America, to Bangladesh, and to Zambia.

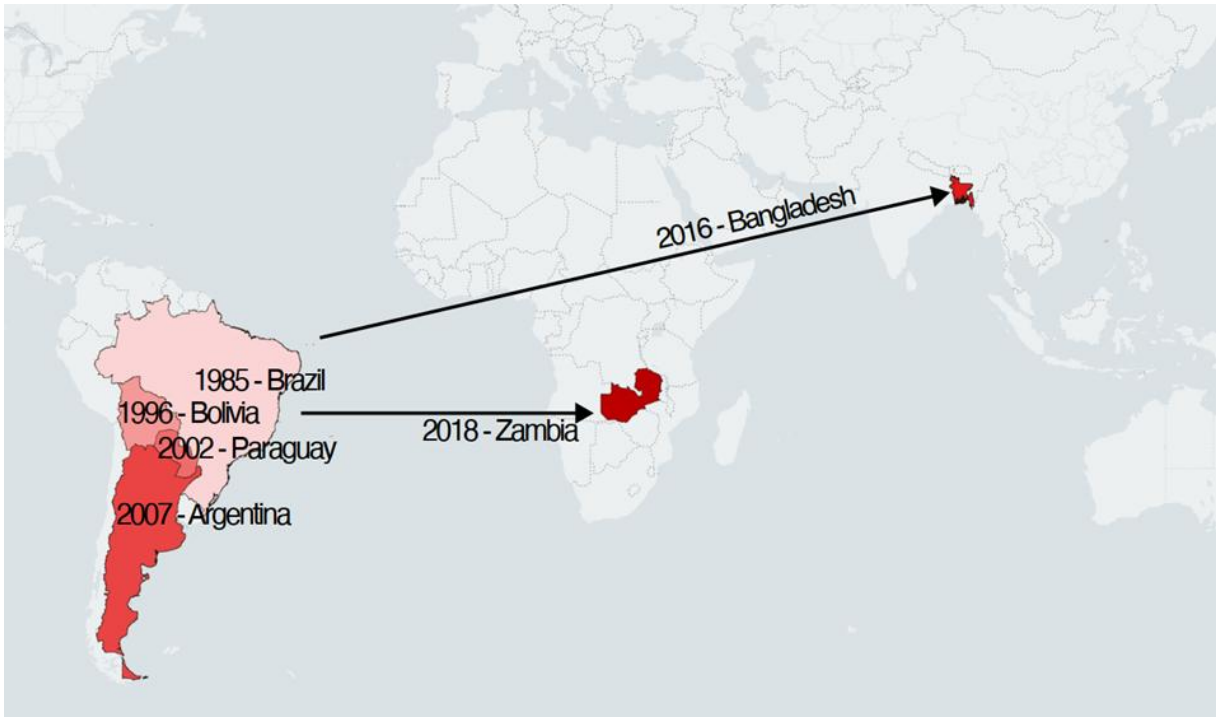

Figure 3. Wheat Blast fungus spread from South America to Bangladesh and Zambia in independent introductions. Figure produced by the author using kepler.gl (https://kepler.gl).

Plant-pathogen interactions

Effectors in blast fungus

Plant pathogens secrete effectors, which are proteins which help them colonise the cells of the host, and contain an amino acid sequence called a signal peptide at the N-terminus (Petit-Houdenot *et al.*, 2020). Signal peptides are short sequences which induce secretion out of the pathogen and into the host cell, and are lost before the protein becomes functional. The host, meanwhile, can utilise Resistance (*R*) genes to attempt to identify effectors, and if they are detected, this leads to a cell death response termed hypersensitive response, and the gene encoding the effector would be referred to as an *AVR* (avirulence) gene (Zhang *et al.*, 2015). This *R* gene is usually an intracellular NLR (Nucleotide-binding Leucine-Rich Repeat) (Jones, Vance and Dangl, 2016). Some interactions involve pairs of NLRs, with a sensor NLR specialised for pathogen detection and a helper NLR which is responsible for signalling and triggering of the response.

MAX effectors (Magnaporthe AVRs and ToxB) are an important category of effectors in Blast fungus, making up between 5-10% of secreted proteins (Petit-Houdenot *et al.*, 2020). As discussed in that study, they have divergent sequences but similar structure, and are present across different host-specific lineages.

The effector PWT3 in isolates infecting on *Lolium* and *Avena* inhibits *M. oryzae* from infecting *Rwt3*-carrying wheat, which is an *R* gene able to detect PWT3 and is present in most wheat grown globally (Inoue *et al.*, 2017). Inoue *et al.* discussed that Anahuac cultivar had become commonly grown in the early 1980s in Brazil due to its suitability for the soil and good yields. They suggested that being widespread and lacking Rwt3 enabled infection of this cultivar by *M. oryzae* despite the expression of the Ao-type *PWT3* effector gene by the pathogen. They also discussed that since blast fungus was successfully growing close to Rwt3-containing cultivars, this led to the loss of functional PWT3 due to selection pressure, and ultimately allowing a new lineage to emerge, the *Triticum* lineage, which was able to infect on a broader range of wheat cultivars.

Previous work (Inoue *et al.*, 2017) discussed that PWT4 is present in *Avena*-infecting and *Lolium* lineage isolates, alongside PWT3, but *Lolium* isolates contain a virulent form of PWT4, whilst the *Avena* form is avirulent on wheat. Further, they experimentally demonstrated that infection of the wheat spike by *Triticum* isolate Br48 transformants with either PWT3 or PWT4 could not successfully infect the *Rwt3* and *Rwt4* carrying Norin 4 wheat cultivar, whilst wild-type Br48 could infect. Meanwhile, only the Br48 transformant containing introduced PWT3 was avirulent on *Rwt3*-carrying Chinese Spring, and only the *PWT4*-carrying transformant was avirulent on the *Rwt4*-carrying Transfed cultivar. They demonstrated that the wild type isolate and both transformants with introduced PWT3 or PWT4 were able to infect the Hope cultivar which lacked both Rwt3 and Rwt4. They also used the *Lolium* isolate TP2 and *Avena* isolate Br58. By disrupting PWT3 in the case of TP2, and PWT3 or PWT4 in the case of Br38 they were able to achieve infection on the Rwt3 or Rwt4 carrying wheat cultivars. They found that disrupting both PWT3 and PWT4 for Br58 resulted in virulence in all four cultivars.

The same study suggested that a *Brachiaria* isolate closely related to Br35 may have contributed a 1.6Mb chromosome region carrying the B type of PWT3 to an ancestor of the *Triticum* Br48 isolate, the isolate used in the infection assays in this work, and as such is able to cause virulence on *Rwt3* and *Rwt4* carrying wheat cultivars (Inoue *et al.*, 2017).

The B71 isolate contains a subtype of PWT3, the Atc type which originated in southern Brazil in the 1990s before spreading more extensively throughout the continent, and then further to become the only form present in the 2016 Bangladeshi outbreak (Inoue *et al.*, 2017). As discussed in that study, the Atc type is characterised by disruption from insertion of Pyret and RETRO5 transposable elements, enabling virulence on *Rwt3* carrying cultivars.

Over 90% of Ethiopian cultivars contain Rwt4, and most South American cultivars contain Rwt4 (Inoue *et al.*, 2017). However, as shown in that work, in the region of India and Bangladesh, it is present in all cultivars. That study does not include data for Zambia, so the situation there is not clear, but since they show that Rwt4 is present in more than 80% of cultivars worldwide, it is likely that many cultivars grown in Zambia may also contain Rwt4.

This demonstrates the importance of PWT3 and PWT4 in determining host range in the *Triticum* and *Lolium* blast fungus lineages as well as the importance of selection of cultivars containing both Rwt3 and Rwt4 to prevent further host jumps.

In the wheat blast fungus, the AVR-Rmg8 effector is detected on attempted infection of plants carrying the *R* gene *Rmg8* (Inoue *et al.*, 2021). The presence of AVR-Rmg8 is of vital importance to the ability to infect, with the B71 lineage containing AVR-Rmg8, whilst other wheat-infecting isolates contain alleles which are partially or completely able to evade immunity (Latorre *et al.*, 2023). PWT4 is able to suppress host resistance to AVR-Rmg8 (Inoue *et al.*, 2021), and PWT4 is not present in the B71 lineage (Latorre *et al.*, 2023).

The impact of blast fungus

Wheat blast causes significant negative impacts on wheat cultivation

The impact of wheat blast fungus being present in a region can be dramatic. It has been suggested that wheat blast fungus was a major reason for the land area in the Brazilian state of Mato Grosso do Sul which had been utilised for wheat production, falling by 95%, between 1987-2016 (Ceresini *et al.*, 2018). However, the degree of reduction in yield can be highly variable, as demonstrated by two wheat cultivars grown in close proximity to each other in Brazil (Sao Paolo State) experiencing yield losses of 32% and 14%, in 2005 (Urashima *et al.*, 2009).

In 2016, in the initial infection season, the yield reduction was nearly 50% in the affected regions in Bangladesh, however, over the next four years the yield loss did not exceed 5% (Singh *et al.*, 2021). That study discussed both changes in management practices and weather patterns as possible causes. In contrast, they mentioned that in the first recorded wheat blast outbreak in Bolivia, in 1996, the yield reduction neared 80%, but increased to almost 100% in the subsequent year, and in following years the amount of wheat grown in the country greatly decreased. It is unclear if this difference reflects changes in management practices as a consequence of lessons learnt from the initial South American spread, resulting in decreased yield losses, or if this is a result of the clonal lineage spread in Bangladesh and Zambia. It is also possible it could be a result of an increased proportion of modern isolates containing mini-chromosomes combined with a potential impact from mini-chromosome presence in an isolate, although the data on mini-chromosome presence over time is incomplete. Such an impact might be caused by the effector composition of the mini-chromosome.

A significant challenge in combatting wheat blast fungus is the shortage of resistant cultivars. As discussed in previous work, the *Triticum* lineage appears to be increasing in aggressiveness over time, with the partial resistance offered by the $2N^VS$ translocation (originating from *Aegilops ventricosa*) being evaded in 2016 in Brazil (Valent *et al.*, 2021). As discussed in that study, previously, the $2N^VS$ region provided the only effective genetic resistance and is widely used in cultivars around the world, but was not usable in all wheat cultivars. A 2020 study found only one cultivar (BARI Gom 33) with partial resistance, out of a total of 16 varieties studied in Bangladesh (Biswas *et al.*, 2020). Further work is ongoing to incorporate additional *R* genes to introduce to $2N^VS$ which could increase robustness in the face of an adapting pathogen (Valent *et al.*, 2021).

A second major issue is the low efficiency of fungicides. Whilst fungicide application has tended to be effective in Bangladesh, its utility in South America varies from cultivar to cultivar, and lacked ability to control the pathogen under suitable environmental conditions for the pathogen (Singh *et al.*, 2021). Overuse of fungicides comes with attendant risks of the development of fungicide resistance, as well as potential harms to health and the broader environment.

If wheat is grown in close proximity to wild grasses, there is a risk of generating crosses with blast strains infecting those grasses which could increase variation. Additionally, the Bangladeshi population may potentially spread further, such as into northern India, where there is a risk of crossing with fertile rice blast isolates, as discussed in another study (Cruz and Valent, 2017). One explanation for the spread of wheat blast fungus into Zambia and Bangladesh has been via the international grain trade, and this could therefore lead to further spread of the pathogen into India, China, the US, Australia, and Europe (Singh *et al.*, 2021). In particular, any spread into India and China could be devastating due to their importance as wheat-growing nations (Islam *et al.*, 2020). As discussed previously (Valent *et al.*, 2021), a climate suitability model shows that conditions in the US may allow for wheat blast fungus infection.

Climate change is likely to significantly alter both methods and patterns of crop growth globally, which may result in further spreading plant pathogens to new areas which may be susceptible to them. It has been suggested that climate change will exacerbate the impact of wheat blast in Bangladesh due to environmental impacts benefiting it (Fones *et al.*, 2020).

Previous work has found that between 1.2% and 3.5% of total blast fungus genes have presence/absence polymorphism between isolates belonging to different host-specific lineages (Yoshida *et al.*, 2016). Some of these may increase their ability to adapt to their environment, or may lead to further specialisation.

Possible management strategies include the use of biocontrol products, expansion of diagnostics to allow earlier detection, integrated management strategies and deployment of novel *R* genes (Islam *et al.*, 2020). Furthermore, collaboration within the community has been highlighted as critical, such as through open science (Kamoun, Talbot and Islam, 2019).

Mechanisms of pathogen adaptation

Generally, a pathogen may adapt to a host through shortening the generation time (time between generations) and increasing the reproductive rate (amount of offspring in a population or by an organism per unit time) in comparison to the host (Croll and McDonald, 2012), whilst large population sizes can help retain genetic diversity. As mentioned by Croll and McDonald, potentially beneficial mutations may arise through genetic recombination (exchange of material between or within chromosomes) or via genetic reassortment (exchange of material between lineages or species). They further discussed that switching of the reproductive mode may be an important feature of emerging epidemics, such as a new clonal lineage arising containing beneficial alleles which can rapidly spread across a host population.

Pathogen evolution and the role of mini-chromosomes

Adaptable genomic compartments are frequently used to aid pathogens in adaptation to the host. For example, the plasmids used by bacteria are known to be able to carry antibiotic resistance factors and toxins. Other examples include satellite RNA in viruses and accessory chromosomes in fungi.

Ectopic recombination, copy number changes and inclusion of genes originating from foreign sources are linked with the rapidly evolving genomic regions in pathogens (Croll and McDonald, 2012). Ectopic recombination is characterised by genetic recombination occurring between homologous sequences at non-allelic sites, therefore leading usually to chromosomal rearrangement (Hartl and Cochrane, 2019). Typically, it is detrimental to the organism if occurring in the core genome, due to the integral functions of many genes in the core genome. However, the more rapidly evolving accessory genome has been identified to tend to feature more mesosyntenic rearrangements than the core genome, featuring particularly many inversions and order alterations (Bertazzoni *et al.*, 2018). This leads to what has been termed a "two-speed genome" (Dong, Raffaele and Kamoun, 2015).

The non-essential mini-chromosomes are genomic compartments which exist separately to the core genome, and can be found in blast fungus genomes (Langner *et al.*, 2021). Despite being known to be present for many years, only recently were mini-chromosomes from blast fungus first sequenced (Peng *et al.*, 2019). A study investigating genomic structural variation in blast fungus isolates infecting on multiple host plants described how mini-chromosomes feature distinctively reduced gene content and greater repeat content than the core genome, they can also undergo structural rearrangements with the core genome (Langner *et al.*, 2021). It has previously been shown that isolates carrying AVRs present on a mini-chromosome which is then lost may regain virulence, as was demonstrated with the rice-infecting isolate 84R-62B where loss of the mini-chromosome carrying two *AVR-Pik* variants yields virulence gain on hosts with *Pik* (Kusaba *et al.*, 2014). *AVR-Pik* is normally present in the core genome in blast fungus (Langner *et al.*, 2021). The FR13 rice-infecting isolate carries those same *AVR-Pik* variants in the mini-chromosome, and also displays indications of structural rearrangements between the core chromosome and mini-chromosome, which may be a mechanism for mini-chromosome origin (Langner *et al.*, 2021).

Previous work (Peng *et al.*, 2019) identified that the B71 mini-chromosome contained elevated proportions of transposons which are typically only frequent at chromosome ends. They suggested that this might indicate a mechanism of material transfer between the core chromosome and mini-chromosome in which transposons play a functional role. The lack of mutations indicative of Repeat Induced Point (RIP) mutation, used defensively against transposons, suggests that many of those transposons have not been inactivated specifically on the mini-chromosome (Peng *et al.*, 2019). They suggested that through such a recombination mechanism, the advantages of the enhanced variation generation of mini-chromosomes could effectively be gained by the core genome through material transferred into the core chromosomes.

Mini-chromosomes may also play a role in transfer of material between lineages which are non-sexually-reproducing and between different species. A previous study (Coleman *et al.*, 2009) identified that accessory chromosomes 14, 15 and 17 in *Nectria haematococcus* differ

significantly from the core genome, and a significant proportion of genes on these chromosomes display greater similarity to *Aspergillus* species sequences than to more closely related *Fusarium* species. As they discussed, that those accessory chromosomes additionally displayed greater gene duplication rates and increased transposon counts also supports inter-species transfer. However, blast fungus isolates from rice-infecting, goosegrass-infecting and foxtail millet-infecting populations have been shown previously to not share the same mini-chromosome (Langner *et al.*, 2021), implying formation or acquisition of mini-chromosomes independently across at least several blast fungus lineages.

Recent work (Barragan *et al.*, 2024) investigated a population of rice-infecting blast fungus isolates present in Italy. The work identified a single unique mini-chromosome called mChrA in AG006 (contig 10), one of these Italian isolates, which highlights mini-chromosome diversity present between closely related isolates. As discussed in that study, mChrA sequence, whilst also present in *Lolium* and *Triticum* isolates, was most similar to sequences in Br62 and B51, both of which are *Eleusine* isolates. Through conducting principal component analysis and constructing phylogenies using SNPs, on just the core genome, the work identified isolates grouping according to their lineages, with *Oryza* and *Eleusine* isolates grouping separately. Meanwhile, conducting this analysis on the mChrA-like sequences yielded altered isolate groupings, in particular featuring *Eleusine* and *Oryza* isolates grouping together (Barragan *et al.*, 2024). Through sequencing and mapping of the Br62 mini-chromosome, obtained through contour-clamped homogenous electric field (CHEF) gel extraction, the study determined that Br62 contained a mini-chromosome highly syntenic to AG006 mChrA, with both exhibiting distinct features compared to their core genomes. It also discovered that in both BR62 and AG006, mChrA carried 7 and 8 likely effectors, respectively. The study found evidence for horizontal transfer of mChrA into the *Oryza* lineage at least nine times during the last 300 years, and based on k-mer sharing patterns between Br62 and AG006 core chromosomes and mChrA, suggested that mChrA was probably introduced into *Oryza* from *Eleusine*. Barragan *et al.* demonstrated horizontal transfer of mini-chromosomes between blast fungus lineages in nature. The work also highlighted the potential role of blast fungus populations infecting on wild grasses to act as a diversity pool able to periodically transfer genetic material into populations infecting on crops which are often grown in close proximity to wild grasses. As such, this provides a mechanism by which clonal populations may acquire new genetic material, enabling adaptation.

Accessory chromosomes can enable effector gain which may alter the ability to infect different hosts, generate new gene functions via duplication of genes or genomic segments (segmental duplication), or via horizontal transfer (Croll and McDonald, 2012). Such chromosomes may also enable gene frequency changes with greater ease, relevant in situations where one pathogen population may be infecting in a region with multiple host cultivars, and such frequency changes may allow different members of the same population to both increase and decrease frequency of a given gene which may be advantageous on one host and deleterious on another.

Muller's ratchet explains how harmful mutations build up in non-recombining populations (Muller, 1964). The lack of recombination prevents elimination of the harmful mutations, under the assumed condition that reverse mutations are uncommon (Gabriel, Lynch and Bürger, 1993). This effect is believed to contribute to an advantage conferred by sexual reproduction over asexual reproduction. Muller's Ratchet is harmful in small populations, which is partly

explained by mutational meltdown. However, horizontal gene transfer may be able to suppress Muller's ratchet in prokaryotic populations, and this effect is indicated to be most protective in multiple separated populations instead of one population, provided there is sufficient horizontal transfer between subpopulations and uptake of extracellular DNA (Takeuchi, Kaneko and Koonin, 2014).

In mutational meltdown, the increase in proportion of detrimental mutations in a small population leads to reduced fitness in the population, causing a cycle of further decreasing population size and an increasing proportion of harmful mutations, which may become fixed via genetic drift (Lynch *et al.*, 1993). Genetic drift is the random change in allele frequency in a population, and tends to remove genetic variation as time progresses due to loss of certain alleles (Reece *et al.*, 2011). It can have significant impact in smaller populations (Reece *et al.*, 2011). Due to mutational meltdown, the population may become non-viable if too large a proportion of the population is unable to reproduce, preventing removal via selection of the accumulated detrimental mutations.

When considering the role of the accessory genome on adaptive evolution of pathogens, the accessory genome is expected to decay over generations due to Muller's Ratchet (Croll and McDonald, 2012). As Croll and McDonald discussed, an example of the degradation taking place may be in *Zymoseptoria tritici*, which displays significant chromosomal length variation in accessory chromosomes. However, large fungal populations may reduce the pace of this process (Croll and McDonald, 2012). As discussed in that work, segmental deletions may be fixed by selective sweeps and the founder effect, due to many accessory chromosome genes not being under a strong selection pressure.

In a selective sweep in a recombining population, a mutation conferring advantage increases in frequency in the population via positive selection, becoming fixed, which consequently decreases the population-wide diversity in the surrounding region as the bordering regions also increase in frequency through genetic draft (also termed genetic hitchhiking) (Nielsen and Slatkin, 2013). Through this process, linked alleles may become fixed (Nielsen and Slatkin, 2013).

The founder effect is the phenomenon of a non-representative, small population becoming isolated from the rest of the population, consequently having low diversity and low standing genetic variation (Reece *et al.*, 2011). Over time, this can lead to speciation. Especially in pest species, a large range with low diversity and small initial population size makes the founder effect important. Depending on the population size involved, the initial establishment of the B71 lineage populations in Zambia and Bangladesh may have been a founder event, which is more likely given that only isolates from the B71 lineage are present in both these regions.

Variation may arise through the comparatively slowly-acting mechanism of mutations, or through standing genetic variation, which is the variation present in large or well-connected populations when the effects of selection pressures are absent. This variation can then lead to adaptation within the population. Both neutral and adaptive genetic diversity are parts of the standing genetic variation, depending on whether the allele in question is currently under selection, but both are important for a population to maintain for long-term population survivability. Standing genetic variation can be an important factor in a young population which

can enable adaptation before mutation has time to act. The founder effect leads to a low standing genetic variation.

Alternative splicing (exons of a gene being combined in different ways to allow one gene to generate multiple proteins) can be important in adaptation and may lead to phenotypic plasticity (where the species can show different phenotypes when present in different environments).

Unless an evolutionary process such as genetic drift, gene flow, the founder effect or natural selection is present, a sufficiently large population will have constant allele frequency across generations, known as the Hardy-Weinberg principle (Reece *et al.*, 2011). Over time, due to the lack of newly-introduced alleles and the predominance of random effects, a population under the influence only of genetic drift experiences a pressure towards genetic uniformity (Reece *et al.*, 2011). This means a gene would usually be either lost or fixed, meaning to become present in 100% of individuals, at which point genetic drift stops, and is only able to occur again if mutation or gene flow lead to development of a new allele. Whilst genetic drift itself may not necessarily result in gene frequency changes benefitting a population, selection will pressure alterations favouring that environment (Reece *et al.*, 2011).

The presence of mini-chromosomes changes over time

A previous study identified that of 14 rice isolates collected prior to 1996, 93% had mini-chromosomes, and wheat-infecting isolates collected from 1988 or earlier did not contain mini-chromosomes (Orbach, 1996). The study discussed that the mini-chromosomes were between 500kb to 2Mb in length. Orbach also reported that the presence/absence polymorphisms in chromosomes were primarily in the mini-chromosomes, and isolates containing mini-chromosomes were more likely to have reduced fertility. Further, the study found an association between infertile rice-infecting isolates and increased numbers of chromosome length polymorphisms, and reported interfertile isolates between different hosts had more stable genome structures. Orbach hypothesised that increased presence of mini-chromosomes in the infertile isolates could either be a driver or result of that lack of fertility.

More recent work (Gyawali *et al.*, 2023) using a Recurrent Neural Network for mini-chromosome detection based on short-read sequence data predicted that a larger set of 196 rice-infecting isolates contained mini-chromosomes in 92% of isolates, which as they note, is similar to values found by (Orbach, 1996). Gyawali *et al.* also found that all wheat-infecting isolates collected later than 2005 had mini-chromosomes, with 92% of all wheat-infecting isolates analysed containing mini-chromosomes. They noted that the three wheat-infecting isolates collected prior to 1991 used in their analysis did not contain mini-chromosomes (BR32, T3 and T25), and that this was supported by the 7 wheat-infecting isolates used in earlier work all lacking mini-chromosomes (Orbach, 1996).

Gyawali *et al.* found that whilst all analysed isolates belonging to the *Lolium* lineage did contain mini-chromosome sequence, *Eleusine* isolates which are less closely related to the *Triticum* lineage frequently did not. They also noted that *Eleusine* isolates tend to be highly fertile, and linked this with the lack of mini-chromosomes. As such, they suggested that mini-chromosome containing wheat-infecting isolates became favoured during the decade following 1990, and highlighted links between mini-chromosome presence and lack of sexual reproduction in those

isolates, suggesting that adaptation to infecting on wheat and *Lolium* coincided with both an increase in asexual reproduction and mini-chromosome presence. Their study used 6 mini-chromosomes and 42 core chromosomes for model training, with the isolates containing mini-chromosomes being B71 (*Triticum* lineage), LpKY97 and TF05 (both *Lolium* lineage), O135 (*Oryza* lineage), and the isolates lacking mini-chromosomes being 70-15 (*Oryza* lineage) and MZ5-1-6 (*Eleusine* lineage). As such, this model suffers from a relatively small amount of training material, and they note a particular bias towards rice-infecting and wheat-infecting isolates. As additional assemblies from multiple lineages become available, this approach will likely become more reliable if based on significantly larger training data sets.

The origins of the Zambian and Bangladeshi wheat-infecting populations

Recent work identified that the Bangladeshi and Zambian wheat-infecting blast fungus outbreaks resulted from separate introduction events (Latorre *et al.*, 2023). As discussed in that study, this most likely occurred coming from South America by isolates which are part of the B71 lineage. It found that Bangladeshi and the more diverse group of blast fungus isolates were distinguishable from each other by a set of 84 SNPs, and those SNPs were further identified to be identical between the Zambian and Bangladeshi isolates, leading to the concept that B71, Bangladeshi and Zambian isolates all formed one lineage, termed the B71 lineage. Further, through PCA clustering with the hamming distance, but also from measuring pairwise linkage disequilibrium, the study determined that these isolates are most likely clonal with a probable origin location for the lineage in South America. That PCA analysis shows that the B71 lineage isolates are highly similar to each other in comparison to the diversity present amongst the South American isolates outside this lineage. Meanwhile, the B71 isolate clusters most closely with Zambian isolates, whilst another South American isolate, 12.1.181 clusters separately from both Zambian and Bangladeshi populations, although closer to the Bangladeshi isolates. The study predicted dates of emergence for the different populations to be 2002-2011 for the B71 lineage, 2010-2015 for the African sub-lineage, and 2009-2012 for the Asian sub-lineage.

The Latorre *et al.* study predicted a significantly quicker evolutionary rate than that previously calculated for rice blast fungus (Gladieux, Ravel, *et al.*, 2018), although during outbreaks this is anticipated. Whilst Latorre *et al.* identified that all 36 studied isolates in the B71 lineage carry *AVR-Rmg8* and therefore trigger immunity in *Rmg8*-carrying wheat, the study highlighted that wheat-infecting isolates outside the B71 lineage can partially or fully infect as they carry one of four virulent alleles of *AVR-Rmg8*. PWT4, which suppresses the resistance response to AVR-Rmg8 conferred by the host carrying *Rmg8*, is not present in the B71 lineage, and infection assays have revealed that *Rmg8*-carrying hosts are not infected by the pandemic clonal lineage isolates (Latorre *et al.*, 2023). As such, the study notes that deploying the *Rmg8* resistance gene may serve as a temporary control measure against the B71 lineage, particularly in countries neighbouring affected regions, but at the risk of that resistance being overcome if deployed in isolation. However, as discussed in other work, Rmg8 (as well as Rmg2, Rmg3 and Rmg7) lose effectiveness when challenged by newer, more aggressive blast fungus isolates, and also at temperatures above 26°C (Valent *et al.*, 2021). The impacts of increasing temperatures resulting from climate change may therefore help defeat previously durable resistance and enable further spread.

As discussed in the Latorre *et al.* study, Strobilurin fungicides are frequently used to combat blast disease. However, as noted in that work, resistance to Strobilurin is relatively common in

the wheat-infecting lineage in South America. They found that thirteen of seventy one isolates of wheat-infecting blast fungus tested had the resistance SNP, but only one isolate from the B71 lineage had it, and therefore the B71 lineage is generally vulnerable to Strobilurin. Despite this positive development for blast fungus management, considerable risk exists that strobilurin-resistance could arise within the Zambian population, confirmed by experimental work that study.

Because the B71 lineage isolates have the MAT1-2 mating type, they are able to sexually reproduce with isolates of the MAT1-1 mating type (Latorre *et al.*, 2023). In Zambia, since African Finger Millet-infecting isolates are present in the vicinity, which have the MAT1-1 mating type, there is the possibility of sexual reproduction between the two which may result in increased virulence in the Zambian wheat-infecting isolates, as discussed by the study. Through experimental work, Latorre *et al.* confirmed both are fertile with each other. Therefore, there is risk for both fungicide resistance and increased virulence in the Zambian population of the B71 lineage, caused either by recombination or mutation.

A recent study (Liu *et al.*, 2023) analysed the B71 lineage, with the aim of using an updated B71 reference assembly they generated to investigate genetic relationships between Zambian, South American and Bangladeshi isolates within the lineage. Their work built upon the previous version of the B71 assembly, achieving telomere-to-telomere chromosomes for all but chromosome 1. Through identifying SNPs within *Triticum* lineage isolates using publicly available data, Liu *et al.* confirmed the results of previous work (Latorre and Burbano, 2021) by concluding that B71, Bangladeshi isolates and Zambian isolates all cluster together, along with the Brazilian isolate 12.1.181, from 2012, which clustered most closely with the Bangladeshi isolates, and that B71 clustered most closely with the Zambian isolates. As such, Liu *et al.* suggested that two sub-branches within the B71 lineage resulted in both outbreaks outside South America. Their identification of 41 SNPs between the Bangladeshi and Zambian populations, located on all chromosomes but which are conserved between Bangladeshi isolates also supports the Latorre et al. 2023 conclusions by implying wheat-infecting blast fungus was introduced to Zambia from South America rather than from Bangladesh.

Mini-chromosome divergence

Liu *et al.* also showed structural variation between mini-chromosomes within the B71 lineage, which was not present between core genomes (Liu *et al.*, 2023). Most of the sequence content of the B71 mini-chromosome was present in all isolates they analysed. Two B71 mini-chromosome segments were not conserved in all Bangladeshi isolates, however, the B71 mini-chromosome sequences were conserved with the Zambian isolates with the exception of a 59kb region missing from ZM2-1 (a Zambian isolate collected in 2018) and ZMW20_04. Their read depth analysis suggested that one mini-chromosome was present across the Bangladeshi isolates they analysed, whilst some of the Zambian isolates carried more than one which were highly similar in sequence, though they cannot be distinguished in genome assemblies. Additionally, through *de novo* genome assembly of three isolates, ZMW18_06, ZMW19_09 and ZMW20_04, they identified a contig with sequence similarity to the B71 mini-chromosome, therefore assigning a mini-chromosome contig.

Previous work highlighted the presence of *BAS1* and *PWL2* in close proximity in the B71 mini-chromosome, whilst in the rice-infecting 70-15 isolate they are located separated on different

core chromosomes (Peng *et al.*, 2019). As discussed in Liu *et al.*, in addition to B71, ZM2-1 carries the *BAS1* and *PWL2* effector genes on a conserved segment of the mini-chromosome which was also present in B71, ZMW18_06, ZMW19_09 and ZMW20_04, and short read data suggested these effectors are present throughout the B71 lineage.

Whilst the core genomes of the Zambian and Bangladeshi populations are highly similar, Liu *et al.* highlighted the importance of the mini-chromosome within the B71 lineage due to their continued presence in the lineage, and suggest they may be conferring advantage (Liu *et al.*, 2023). They further point to the prevalence of structural variation and increased repeat content occurring in mini-chromosome contigs within the B71 lineage, compared with the core genome, along with the conservation of genes such as *PWL2* and *BAS1* as evidence of rapid adaptation and a potential role in enabling greater virulence. They also note that the increased co-occurrence of *PWL2* and *BAS1* throughout time in wheat-infecting isolates could indicate that mini-chromosomes have become more common over time, due to both effector genes being carried on the mini-chromosome.

Supplemental data from Liu *et al.* demonstrated whole-genome and core-genome SNPs were most frequent between Zambian and Bangladeshi isolates, but when considering only SNPs on the mini-chromosome, there were particularly few between Zambian isolates and other Zambian isolates, or between Zambian isolates and B71 (Liu *et al.*, 2023). However, that study found that many SNPs were present on the mini-chromosome between Bangladeshi isolate mini-chromosomes and any other mini-chromosome, including other Bangladeshi isolates. From Liu *et al.* it appears that the diversity in mini-chromosome sequence is more divergent within the Bangladeshi population than it is within the Zambian population, whilst the core genome does not appear to show this same pattern. Given that the Bangladeshi population is the older one, this pattern is to be expected. The reference used by Liu *et al.* was B71, which may distort the magnitude of this result due to being more closely related to the Zambian population than the Bangladeshi isolates.

Recent work (Kobayashi *et al.*, 2023) investigated epigenetic modification in the wheat-infecting isolate Br48, concluding that the H3K27Me3 mark represses genes derived from horizontal transfer as a defence mechanism. They suggested that some of these genes developed to be expressed during infection and could possibly benefit the pathogen in its niche. Whilst their work promotes epigenetic modification as an important mechanism in blast fungus for domestication of horizontally-transferred genes, further systematic analysis across lineages and gene sets would be beneficial.

Methods of wheat blast control

Efforts to control wheat blast have been hampered by the few resistant cultivars available. Analysis of sixteen wheat cultivars growing in Bangladesh found that the sole example displaying any resistance (BARI Gom 33) only had partial resistance (Biswas *et al.*, 2020). They suggested that BARI Gom 33 should be either used as a source of material for further development of resistant wheat cultivars, or should itself be grown directly in affected regions.

It has been demonstrated that wheat seedlings (Anahuac cultivar) grown from infected seed can themselves have infected primary leaves (Goulart and Paiva, 1990), meaning that it is feasible for long-distance spread of wheat blast to occur via movement of infected seeds (Cruz

and Valent, 2017). However, development of blast fungus symptoms from infected seeds to the 15-day old seedling has been demonstrated in the previously mentioned infected-seed transmission study to be inhibited by application of the fungicides Thiram and Iprodione (Goulart and Paiva, 1990).

Precisely timing fungicide spraying, seed treatment and ensuring usage of resistant cultivars were successfully used in an integrated disease management system by Brazil in the 1980s to reduce losses (Mehta *et al.*, 1992), whilst multiple countries in South America have also increased production by delaying planting (Coelho *et al.*, 2016). It is believed that this delayed planting can succeed due to preventing the environmental conditions most suitable for blast fungus happening at the same time as wheat heading (Cruz and Valent, 2017).

Whilst fungicide application in South America has been found to offer some defence against blast fungus, application to seeds has been most effective (Cruz and Valent, 2017). As mentioned previously, strobilurin fungicides, used frequently to control wheat blast, may be used to control the B71 clonal lineage, whilst the R gene *Rmg8* can serve as a control method due to *AVR-Rmg8* being carried throughout the B71 lineage (Latorre *et al.*, 2023). Therefore, as discussed in Latorre *et al.*, *Rmg8*-carrying cultivars have been suggested to be grown in regions surrounding countries with blast fungus presence, however, both these strategies are temporary solutions due to the expectation that either mutation, or genomic recombination with *Magnaporthe oryzae* isolates found in the region but belonging to other lineages, may allow emergence of isolates able to resist fungicides, or isolates that display greater virulence. In particular, Latorre *et al.* highlighted these risks with experimental work demonstrating that Zambian blast fungus isolates can gain Strobilurin resistance and have fertility with opposite mating type African Finger Millet isolates.

Replacing cultivation of wheat with a different crop for several years, known as a wheat holiday, has also been suggested, but can be politically impossible due to the increased dependence on importation of wheat (Mottaleb *et al.*, 2019). As was also discussed in previous work, whilst wheat holidays were employed both within Bangladesh and on the Indian side of the Bangladesh-India border, the *Triticum* lineage is still causing infection both on wheat and triticale (Valent *et al.*, 2021).

Summary

The highly damaging wheat-infecting lineage of *Magnaporthe oryzae* has spread from being present only in South America, to also establishing itself within Asia and Africa. In Bangladesh, it was first identified in 2016, whilst it is believed to have first infected in Zambia in the 2017-2018 season. The group of isolates responsible for this spread belongs to a single, clonal lineage, called the B71 lineage. Effector presence/absence polymorphisms between different blast fungus isolates can enable or inhibit infection on different host species or cultivars, and as such can indicate pathogen adaptation.

In this work, I want to show my attempt at identifying how the presence of population-specific effector candidates within different populations in a clonal lineage of *Magnaporthe oryzae* might demonstrate signs of ongoing adaptation to the new environment and hosts these populations may be challenged by. In Chapter 2, I will outline my methods, accompanied by the code in Appendix 1. Chapter 3 discusses the genome structures and genome rearrangements of the

B71 lineage isolates, followed by the identification of population-specific effector candidates. Next, Chapter 4 details how I identified the distribution of one of those candidate genes, *Art1_WB_ZM,* throughout multiple blast fungus lineages, particularly in the Zambian population of the B71 lineage. I then generated effector-deletion transformants before performing infection assays to attempt to identify any virulence changes when infecting on wheat cultivars. In Chapter 5, I detail a similar process, but focussing on the *APiasL3* candidate present in the Bangladeshi members of the B71 lineage. In addition to attempting to generate effector-deletion transformants, I also investigated patterns of residue conservation and indications of selection on certain residues, through analysing *APiasL3* together with the other members of the gene family it is a part of.

Chapter 2: Methods and materials

Genome sequences

Genome assemblies for three Bangladeshi isolates (BTJP4-1, 58 contigs; BTGP1-b, 73 contigs; BTMP-S13-1, 15 contigs) were obtained from the Kamoun lab based on previous projects within the group. Twenty additional sequenced genomes from Bangladeshi isolates and thirteen from Zambian isolates were obtained through the Kamoun lab to be used in the analysis. Additional genome sequences used to assess presence/absence of effector candidates throughout blast fungus lineages were provided by Thorsten Langner and Vincent Were.

Identification of mini-chromosome contigs

In work carried out during my predoctoral internship within the Kamoun group, I used existing data (produced by Thorsten Langner and Adeline Harant) consisting of sequenced reads from extracted mini-chromosomes obtained using contour-clamped homogenous electric field (CHEF) gels, from three Bangladeshi isolates (BTJP4-1, BTGP1-b, BTMP-S13-1). I mapped these reads against each isolate's genome assembly to attempt to identify mini-chromosome contigs (Langner *et al.*, 2021). Using the method detailed in Appendix 1, I used BWA (version 0.7.17), samtools (version 1.3.1) and bedtools (version 2.20.1) to map mini-chromosome reads onto the respective genome assemblies, and displayed them using the Circlize library for R, splitting them into difference plots at the contig size thresholds of 200kb and 2Mb. Mapping was also performed for nanopore reads, and RepeatMasker was used in conjunction with a B71 repeat library from (Peng *et al.*, 2019), and these were used to check for assembly errors and possible artifacts in the predicted mini-chromosome contigs. If the measured mini-chromosome read coverage of a contig significantly exceeded the background level, then it was recorded as a mini-chromosome contig.

Computational extraction of isolate-specific genomic regions

A custom pipeline (the code is provided in Appendix 1) was developed with Python and Bash, utilising the Nucmer alignment tool from the MUMmer (version 3.1) package (Marçais *et al.*, 2018), to extract isolate-specific genomic regions. With this approach each Bangladeshi and Zambian genome assembly was aligned to B71, and only if there was no alignment greater than 1kb between the isolate and the B71 reference would genomic segments be considered unique. Nucmer (from MUMmer version 3.1), Samtools (version 1.3.1) and Bedtools (version 2.29.2) were used in this method. A schematic of the method is shown in Figure 4.

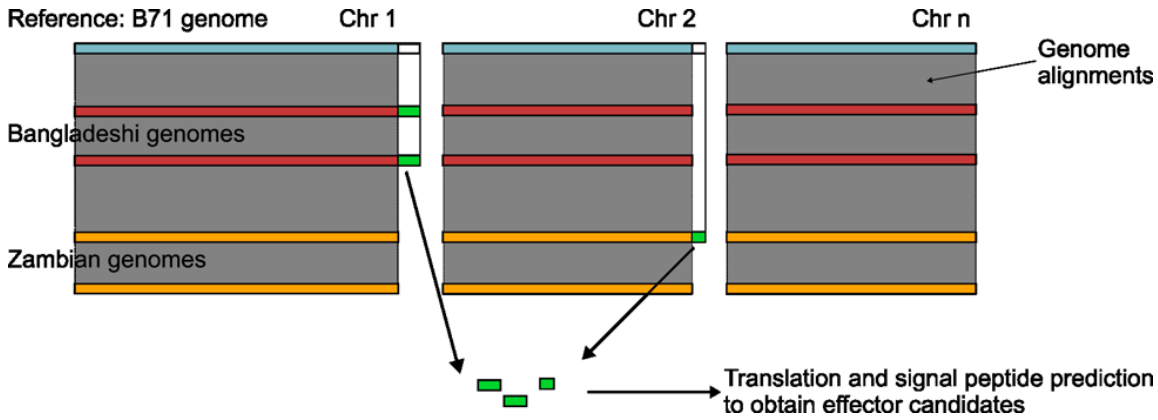

Figure 4. Outline of isolate-specific genomic region extraction method, utilising pairwise Nucmer alignment of isolates from Zambian and Bangladeshi populations to the B71 assembly.

Effector candidate identification

With the aim of identifying population-specific effector candidates absent in B71, I gathered those isolate-specific genomic regions extracted for each Bangladeshi and Zambian isolate. The original dataset used for this was:

- 3 Bangladeshi assemblies (mean of 49 contigs)
- 13 Zambian assemblies (mean of 129 contigs)
- 20 additional Bangladeshi assemblies (most with 1000-2000 contigs)

A further custom pipeline which was developed and run by Joe Win, using established methods (Torto *et al.*, 2003; Raffaele *et al.*, 2010) but with the modification of not using WolfPSort, used the assemblies as an input for effector candidate screening using 6-frame translation, SignalP, THMHH, TargetP, outputting a set of predicted secreted proteins.

I developed a pipeline (code provided in Appendix 1) for gene distribution identification over a group of lineages, using Bash, Python 3.7.6 and terminal-based BLAST software (2.10.1) (Camacho *et al.*, 2009) for control and match identification, as well as the Seaborn Python library for visualisation. This pipeline was used to identify the distribution of effector-candidates across multiple lineages of blast fungus. The cladogram used for the visualisation was generated using the kSNP3 software (Gardner, Slezak and Hall, 2015). As in Gladieux *et al.* (2018), *Digitaria* and *Pennisetum* isolates served as an outgroup.

A further development of this pipeline was the use of recursive search, where the results from an initial search were used as queries for a secondary search. This was used in particular for identifying APiasL3 family members.

Blast fungus samples

Blast fungus samples obtained from the Kamoun lab had originally been collected in Bangladesh in 2016 (BTJP4-1 and BTMP-S13-1) during the first infection season, and in 2017 (BTGP1-b). These samples had been maintained previously in the lab and long-term stocks generated through growth on complete medium (CM) agar plates, and allowing a colony to grow over filter paper squares which were then desiccated, as described in previous work (Molinari and Talbot, 2022). Instead of using a vacuum desiccator for 48 hours to dry the

samples, as mentioned by Molinari and Talbot, I instead dried the samples for at least 10 days in a sealed box containing large quantities of silica gel desiccant beads. The *M. oryzae* samples were stored in a freezer for long-term storage. Zambian blast fungus isolates were provided by Batiseba Tembo. Additional samples were provided by Vincent Were.

When growing material for infection assays, samples were grown on CM agar plates, using methods described in (Molinari and Talbot, 2022). All operations were performed in a laminar flow hood with flame-sterilised tools. These new *M. oryzae* cultures could be started either from a dried filter paper from a long term stock, or from a small sample of mycelium extracted from an existing growing colony. In the latter case it is important not to continuously generate new plates from previously growing samples, so no more than 3 generations of plates were produced before starting afresh with a new dry filter paper stock sample. Colonies growing on plates were typically harvested after 7-10 days. Samples used for generating material for transformation were initially extracted from a colony growing on a CM plate, and then grown in liquid CM broth placed on a rotary shaker, as described in the below section on transformant generation. Both in the cases of growing *M. oryzae* on agar plates and in liquid medium, the cultures were incubated at 25°C with a 12-hour photoperiod.

Effector-deletion transformant generation

In order to generate effector deletion transformants, I used the split hygromycin system proposed for use in fungi (Goswami, 2012) to perform targeted gene replacement of the effector candidate with the hygromycin B phosphotransferase gene, which encodes resistance to the antibiotic hygromycin B.

Using PCR, I amplified the 1kb upstream and downstream regions either side of the gene of interest (using primers oAM1 paired with oAM2, and oAM3 paired with pAM4, as listed in the table of primers in Appendix 2), then performed a fusion PCR with primers inside the hygromycin-cassette region, forming two cassettes with complementing segments of the Hygromycin marker gene fused either to the upstream or downstream flank. I ran a gel to purify the construct by extracting the bands corresponding to these fused products using the Machery-Nagel PCR clean-up and Gel extraction kit, before extracting the DNA. Both cassettes were then used in the transformation. During preparation for transformant generation where PCR products were required for use in downstream stages, Phusion DNA polymerase was used during PCR. I used 34 cycles with a denaturation time of 15 seconds at 98°C, and annealing time of 25 seconds at 63°C, and a 60 second extension time at 72°C. For most diagnostic purposes throughout the project where PCR products were not used in downstream stages, DreamTaq DNA polymerase was used instead. For this I used 34 cycles with a denaturation time of 20 seconds at 95°C, an annealing time of 20 seconds at 63°C, and a 60 second extension time at 72°C.

To proceed with the transformation of *Magnaporthe* oryzae, I followed the method described at https://dx.doi.org/10.17504/protocols.io.kxygx7pokl8j/v1 (Haeussler and Langner, 2025).Cultures of ZMW18_10 were grown on CM plates as described in the previous section. Rather than blending the extracted mycelium used to inoculate the broth with a two-speed blender as described by (Molinari and Talbot, 2022), small, finely-cut cubes of the agar plate were extracted only from the outer edges of the colony. These were then used to inoculate a liquid culture in 250ml liquid CM broth, incubated at 25°C for 48 hours on a rotary shaker with

a 12-hour photoperiod. I prepared protoplasts from this culture by filtering the culture with sterile Miracloth, transferring the mycelia into a mix of 40ml 0.7M NaCl solution and 400mg lysing enzymes from *Trichoderma harzianum* and leaving it for 2 till 3 hours on a gently rocking platform. This was filtered through sterile Miracloth and then centrifuged at 4,000rpm at 4°C for 10 minutes, resuspended in 50ml STC before centrifugation again for 10 minutes. I then mixed the protoplasts with the purified cassettes which were at least 100ng/µl, with a total volume of between 100µl and 200µl STC, and left it on ice for 15-25 minutes. I added 1ml of PTC, and incubated for a further 15 to 20 minutes. I added 10ml OCM (liquid osmotically stabilised CM), covered the samples with aluminium foil to prevent light exposure, and left them overnight on an oscillating shaker at 75rpm at room temperature.

The next day, I mixed the protoplasts with OCM agar medium, being careful to mix the fragile protoplasts with the reheated OCM medium only when the OCM temperature dropped to approximately 45°C. Higher temperatures risk harming the protoplasts, whilst at lower temperatures the OCM will solidify before it can be poured in plates, and it cannot be reheated for repouring once it contains the protoplasts. I poured this mixture in a petri dish with a thin top layer of Hygromycin (500µg/ml) mixed with CM medium (approximately 15ml per plate), which was incubated overnight. Colonies which grew up through the medium and emerged at the top of the plate should be at least partially resistant to Hygromycin, indicative of successful integration of Hygromycin-resistance, and so after 6 days, material could be extracted from colonies growing on top of the medium using a pipette tip, and grown on a second plate containing CM with Hygromycin to validate this. Any transformants able to grow on this second Hygromycin-containing plate were subcultured to generate material for long-term storage and for further analysis using the technique described in the previous section.

To initially screen transformants, I tested if the 1kb upstream region and 1kb downstream region before and after the effector candidate had fused to the Hygromycin-resistance gene and that both flanks had integrated into the genome in the expected location. To do this, I extracted samples from the Hygromycin selection plates used to screen the transformants, and using the Phire Plant Direct PCR kit, ground the samples before performing PCR using Phire Hot Start II DNA Polymerase with diagnostic primers (designed to start and end on either side of the 1kb flanks; primer pairs d1-d2 and d3-d4 for each effector candidate) to amplify the upstream and downstream flanks. For the PCR, I used 34 cycles with a denaturing time of 20 seconds at 95°C, an annealing time of 20 seconds at 63°C, and an elongation time of 60 seconds at 72°C. With this method I anticipated bands of the appropriate sizes only if the regions had correctly been integrated. Primers used as a control corresponded to a sequence known to be present in the BTJP4-1 mini-chromosome but absent in the Bangladeshi isolate core genomes, and were provided by Thorsten Langner.

To identify additional integrations of the Hygromycin resistance gene elsewhere in the genome specifically in the Bangladeshi transformants, I attempted to use Southern blotting to identify its gene sequence within the transformant genome. The primers used for this are listed in the table of primers in Appendix 2, named oAM1 and oAM4, which correspond to the whole of the upstream and downstream flanks of the region surrounding the effector candidate, and the transformation site. Restriction enzymes were selected which would cut upstream of the upstream flank (EcoRV), and downstream of the downstream flank (EcoRV), and within the Hygromycin cassette (NotI), but not within the effector candidate gene. Through this method, the DNA sample (which I extracted using a CTAB-based protocol) undergoes DNA digestion

using restriction enzymes, followed by fragment separation by length via gel electrophoresis. To obtain only single-stranded DNA strands, the sample is denatured using NaOH. Then, after transfer to a filter paper, a labelled complementary strand is added to allow visualisation. If the sequence of interest is present in the sample, a corresponding labelled band should appear. Visualisation techniques may use a radioactive tag, but here the non-radioactive DIG (Digoxigenin) system was used. As such, this method should allow determination of whether the transformation took place successfully and placed the Hygromycin resistance gene between the upstream and downstream flanks, or if the effector candidate is still present between the upstream and downstream flanks.

A second method I used to attempt to identify successful transformants in both the Zambian and Bangladeshi candidate experiments was the use of Illumina whole genome sequencing by Novogene. This was done using the PE150 platform, and the Illumina NovoSeq 6000 machine generating 150bp paired-end reads. The quality score distribution along reads for all samples was above 30, indicative of a low risk of incorrect base call, with Q30% values for all samples being over 92. All samples had a base error rate of 0.03%. Using BWA (version 0.7.17), I mapped the resulting reads against the wild type genome assembly of the isolate used to generate the transformant, as well as a simulated transformant genome where the target effector sequence was replaced the hygromycin resistance gene sequence. This simulated replacement was performed in the Geneious software (version 2023.2.1). I made comparison of read depths for both these cases to confirm success or failure of transformation, which was performed by displaying the mapped reads from BWA in the Integrative Genomics Viewer (version 2.17.0) and measuring the comparative levels.

Plant cultivar sample origin

Seeds from 10 cultivars of wheat and 2 cultivars of barley were sourced from within the Kamoun lab. The wheat cultivars used were:

- Mace
- Julius
- Stanley
- Lancer
- Landmark
- ArinaLrFor
- Jagger
- SY-Mattis
- Fielder
- Chinese Spring

The barley cultivars used were:

- Golden Promise
- Nigrate

Infection assays

In order to check for an impact on virulence due to effector loss in the previously generated transformants, I performed infection assays. Leaf segments (second leaves were used) from

plants which had grown for 10 days in a controlled environment cabinet since sowing were harvested and fastened into enclosable dishes which contained either 4% water agar or damp paper to create a moist environment preventing drying of the leaves. In order to attempt stable placement of droplets, it was important to flatten and straighten the leaves as much as possible by creating a slight tension between two pieces of tape at either end of the segment.

Blast fungus cultures were initiated in parallel to growing the seedlings by extracting a sample from an *M.* oryzae colony already growing on an existing CM plate, and this was placed on a new CM plate and grown for 10 days prior to inoculation. Spores from each *Magnaporthe oryzae* culture were harvested in a laminar flow hood by running water sequentially over each plate of that isolate and mechanically disturbing the culture surface using a sterile bacterial spreader. The resulting liquid was filtered through a 70µm sieve, then centrifuged at 2500 rpm for 15 minutes, and the supernatant removed. The spores were then resuspended in 0.5ml gelatine (adjusted depending on the number of droplets used in each experiment), and the spore concentration determined through microscopy of a 1/10 dilution to obtain a final concentration of 10,000 spores/ml. Droplets (10µl) of spore solution for each isolate were carefully deposited on the surface of each leaf, alternating placement so that each isolate was placed in each possible position on each cultivar, to control for positional variability. A negative control droplet (labelled 'water' in figures) containing only gelatine and no spores was also used. The plates were stored in a growth chamber used for *M. oryzae* growth, and then after 4-5 days were imaged.

Agroinfiltration

The method used in the agroinfiltration experiment follows previous work (Bos *et al.*, 2006; Madhuprakash *et al.*, 2024), but with some modifications, and is summarised below.

Wild-type *N. benthamiana* were grown in a controlled environment chamber and used in the experiment when 4 weeks old. I prepared the Art1_WB_ZM-containing plasmid pAM8, which was designed for use in the infection assay (Golden Gate; components are listed in Appendix 2), through an *E. coli* transformation with a plasmid containing the Art1_WB_ZM sequence and 4 Golden Gate modules, with the transformants grown on a Carbenicillin-containing plate. I used the QIAprep Spin Mini Kit for plasmid purification, and then sequencing was performed via the GeneWiz Sanger sequencing service for the pAM8 plasmid to verify correct module integration.

Two days prior to agroinfiltration, two individual colonies were harvested from a plate containing *Agrobacterium tumefaciens* transformed with the pAM8 plasmid, and used to create two liquid cultures containing antibiotics. These were then placed into a shaking incubator overnight. One day prior to agroinfiltration, a sample of the previous liquid culture was mixed with a new liquid medium, for use in agroinfiltration on the following day at the correct optical density for standardisation.

Infiltration buffer was prepared using an MES buffer composed of 10mM MES and 10mM $MgCl_2$, mixed with 200µM Acetosyringone. The OD_{600} was adjusted using infiltration buffer to 0.6 for both avirulence genes, 0.6 for pAM8, and 0.3 for Pikm.

Leaves were infiltrated with constructs using a syringe without a needle and with gentle pressure, then left and observed for the appearance of infection symptoms.

The experimental setup for each leaf used AVR-PikD expressed with Pikm as a positive control, and AVR-PikC expressed with Pikm as a negative control. Blast fungus infection by isolates carrying AVR-PikD is known to lead to immune response in Pikm-carrying rice lines, whilst AVR-PikC does not (Kanzaki *et al.*, 2012). The strains carrying AVR-PikC, AVR-PikD and Pikm were provided by Jiorgos Kourelis. Infiltration with the pAM8 plasmid occurred in two locations on each leaf.

Chapter 3: Discovery of presence-absence polymorphism of two effector candidates in pandemic lineage

Introduction

Plant pathogens may adapt to better enable them to infect their host, or expand their host range, by gain, loss or mutation of effectors. As discussed in chapter 1, genomic rearrangements, potentially involving mini-chromosomes, may facilitate these developments and can be especially important in asexually-reproducing organisms. As such, any identified presence/absence polymorphism in effector composition between different subpopulations may be indicative of adaptation to their host and environment, and are promising candidates for further study. A greater understanding of effector composition may allow for more effective use of management strategies to prevent disease spread, relevant to the local conditions.

This chapter compares the genome structures of B71, Bangladeshi isolates and Zambian isolates, before discussing the process of identifying mini-chromosome contigs within isolates from Bangladesh. It then compares in more detail genomic structural changes between the Bangladeshi BTMP-S13-1 isolate and B71. This is followed by discussion of the selection of isolates used to identify population-specific genomic regions, and then by the identification of effector candidates within these regions.

Aims

In this chapter, I aim to demonstrate the discovery of effector candidates within the B71 lineage populations with a distinct, population-specific distribution amongst the blast fungus population. The distribution pattern appears to be linked to genomic rearrangements which are specific to either the Bangladeshi or Zambian populations, and may play a role in adaptation. To do this, I first used genome structures and structural variation present between isolates of the clonal pandemic lineage to illustrate genomic changes across the pandemic clonal lineage. Since mini-chromosomes may be important in such rearrangements, I identified mini-chromosome contigs present in the Bangladeshi isolates through utilising sequencing data. To identify genomic regions which may host population-specific effectors, I used a custom pipeline built in Bash and Python, and utilising BLAST, to identify genomic regions which are present in either the Bangladeshi or Zambian but not in the South American B71 isolate. Utilising the identified population-specific genomic regions, I could then predict effector candidates, of which two were selected for further study, as detailed in chapters 4 and 5.

Specific methods

As part of the mini-chromosome identification process, I developed custom R scripts guided and inspired by previous work within the Kamoun lab, for displaying reads from mini-chromosome sequence data, nanopore sequence data and repeat sequence content, on a circos-style plot using the Circlize library. I also developed my own scripts for linear read-

mapping plots and genome alignment plots in Python. The scripts for this work are provided in Appendix 1.

The collection dates of the samples from Bangladesh and Zambia which were used in this work are shown in Figure 5, along with the dates of blast arrival in each region.

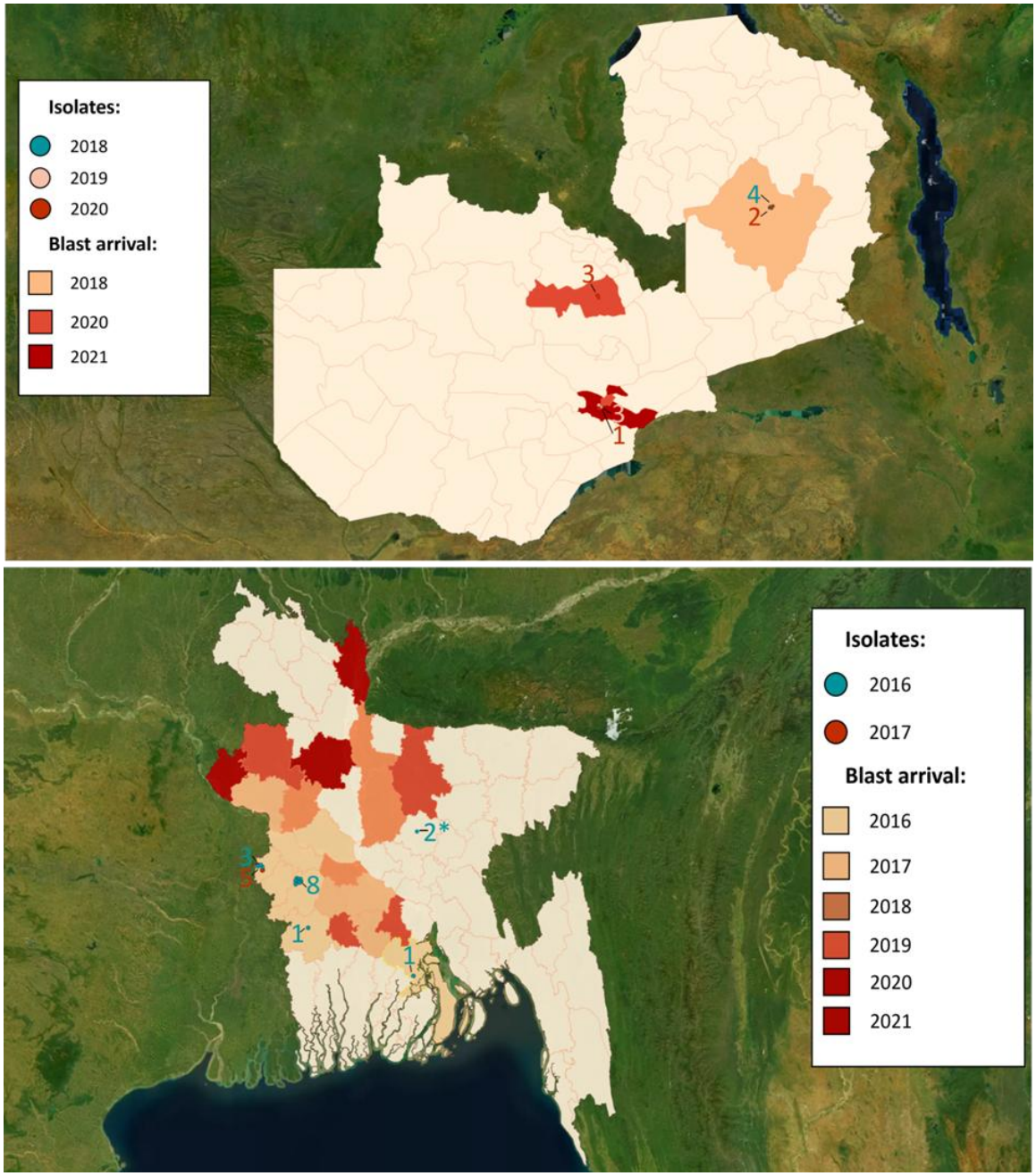

Figure 5. Blast fungus arrival in Bangladesh (bottom) and Zambia (top), along with collection dates for isolates used in this work. The number of samples from each region are marked next to the dots marking their positions, with the colour of the text representing the collection year in the same way as the dots. In the map of Bangladesh, the two samples (BTBa-B1 and BTBa-B2) located in Gazipur (marked with an '') were collected from BSMRAU campus from Barley leaf. Figure produced by the author using kepler.gl (https://kepler.gl).*

Results

Structural variation and the genome structure of Zambian and Bangladeshi blast fungus isolates

I used the software Mauve (using the function Mauve Contig Mover) to reorder contigs in both the Zambian and Bangladeshi assemblies, using B71 as the common reference for all isolates. There were 13 Zambian isolate assemblies and 3 Bangladeshi isolate assemblies used at this stage.

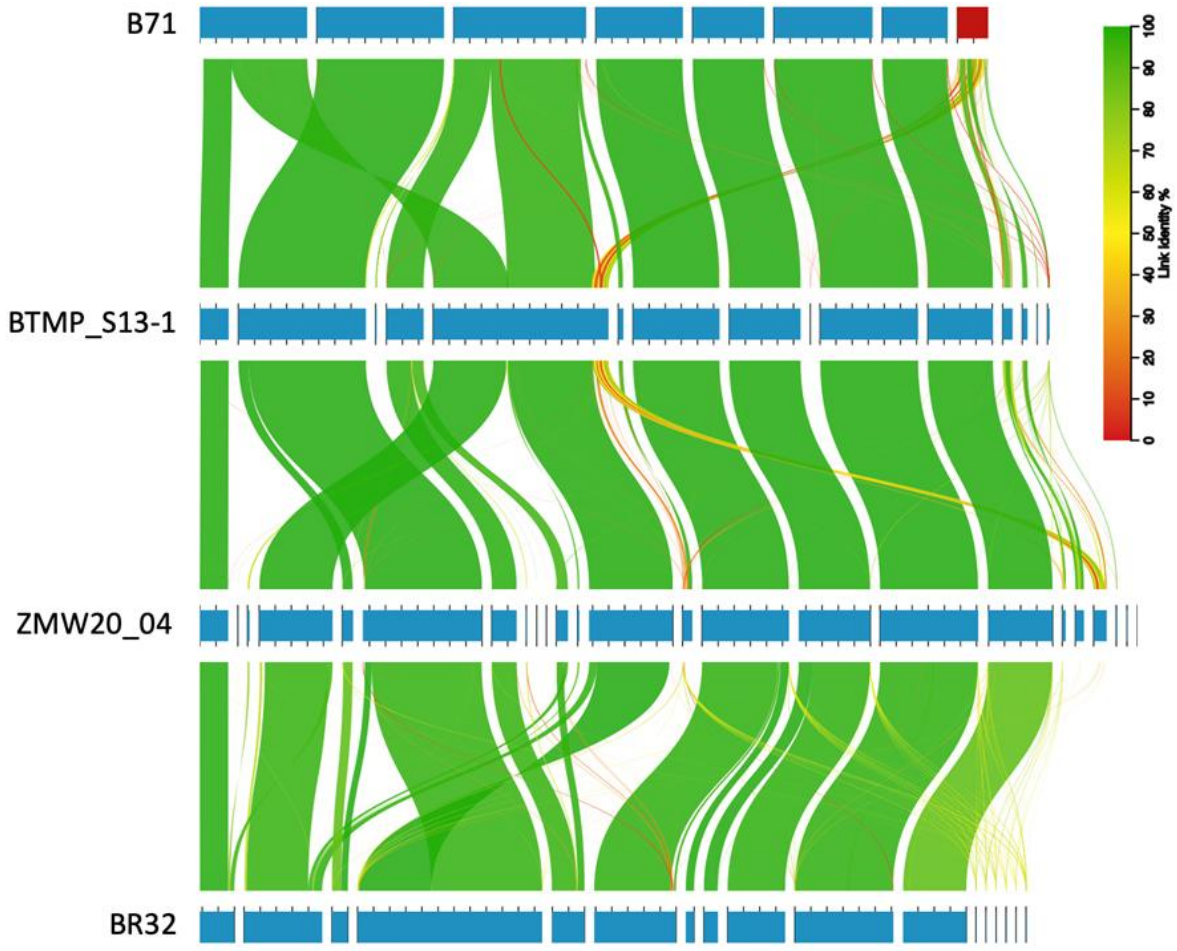

Figure 6. Genome alignments between B71 and members of the Bangladeshi and Zambian populations reveal conservation of mini-chromosome sequence (indicated by the red contig in B71) which is absent in the older BR32 isolate from South America. Figure produced with AliTV 1.0.6.

B71, the South American isolate which was used as a reference for this work due to its relatedness to the Zambian and Bangladeshi isolates and its high degree of sequence conservation with those isolates, has been identified to have 7 core chromosomes and 1 mini-chromosome (Peng *et al.*, 2019). As can be seen in Figure 6, most of the genome of the isolates within the B71 lineage aligns very well, with most of the core genome being shared between B71, and isolates from both the Bangladeshi and Zambian populations. However, several of the core chromosomes in B71 appear to have undergone rearrangements in the Bangladeshi and Zambian populations, such as an apparent splitting of B71 chromosome 1 into two fragments, the initial portion of which is shown to comprise a separate small contig within many of the Bangladeshi and Zambian isolates.

Sizeable segments of the B71 mini-chromosome sequence appear to be integrated into the core genome of the Bangladeshi BTMP_S13_1 isolate, whilst in other Bangladeshi isolates there is no evidence to support integration. Due to this, isolates from the B71 lineage may plausibly have undergone structural rearrangements between the mini-chromosome and core genome. The Brazilian BR32 isolate from 1991, however, lacks a mini-chromosome (Langner *et al.*, 2021) and also does not carry significant lengths of the B71 mini-chromosome sequence in its core genome, as shown in Figure 6. Another isolate which lacks a mini-chromosome is the Brazilian T25 isolate, which was collected in 1988. Neither BR32 nor T25 are in the B71 lineage, and T25 is considered to be a generally less aggressive isolate than more recently collected wheat blast fungus isolates (Peng *et al.*, 2019).

The BR32 isolate has a short contig aligning to the initial segment of B71 chromosome 1, although with a short region fused onto its end, which may suggest that this shorter chromosome is the original form, which only became fused into a much larger chromosome in B71 but remained separated in other isolates within the B71 lineage. The BR32 contigs which align with B71 chromosomes 6 and 7 do not feature any large scale rearrangements, whilst those aligning to B71 chromosomes 4 and 5 feature some smaller contig breaks in BR32 which may be explained by errors during the genome assembly process. It is B71 chromosomes 1, 2 and 3 which feature significant genome rearrangements both between other B71 lineage isolates, and with the older BR32 isolate from outside the lineage.

As an illustration of the broader genomic landscape amongst B71 lineage isolates, in Figure 7, the many rearrangements caused by contig breaks can be seen, which greatly complicates downstream analysis and interpretation. In general, it can be noted that it is a common feature of the assemblies analysed that many within the B71 lineage have a large number of highly-fragmented contigs which frequently align to the B71 mini-chromosome sequence, highlighted with red in the figure.

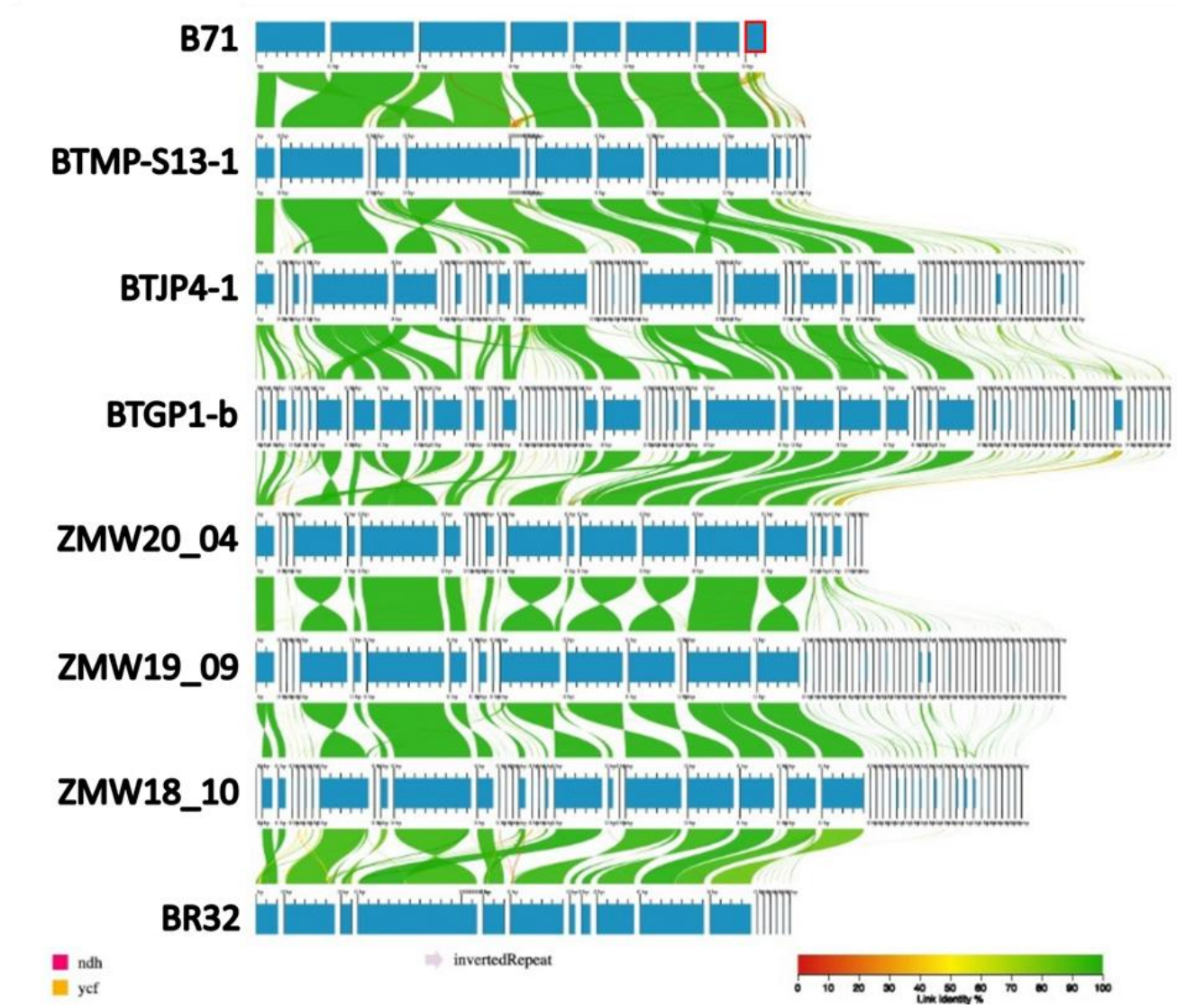

Figure 7. Genome alignments between B71 and members of the Bangladeshi and Zambian populations reveal significant sequence conservation, rearrangements and conservation of mini-chromosome sequence which is absent in the older BR32 isolate from South America. The B71 mini-chromosome contig has been highlighted in red. Figure produced with AliTV 1.0.6 and manually edited for clarity.

Mini-chromosome contigs were identified in Bangladeshi isolates

Following the previous sequencing of mini-chromosomes in the Bangladeshi isolates within the Kamoun lab by Thorsten Langner and Adeline Harant, I used BWA to map mini-chromosome reads onto the genome assemblies. Those contigs having significant coverage were preliminarily categorised as belonging to the mini-chromosome as opposed to the core genome. Through analysing the repeat content and nanopore reads, I verified this assignment by checking for possible signals of assembly errors such as abnormal jumps in nanopore coverage, which could indicate a duplication. A table showing contigs designated as mini-chromosome contigs is provided in Appendix 2.

As can be seen in Figure 8, for the BTMP_S13-1 isolate, three mini-chromosome contigs were identified (see tracks with red fill, and a red dot marking the contig designated as a mini-chromosome contig), all smaller than 2Mb in length. Each displayed consistently high mini-chromosome read coverage for three different CHEF gel extracted mini-chromosome samples.

Further, no abnormalities in nanopore read coverage (tracks with green fill) were detected in these regions, reducing concern about potential assembly errors within these areas. Repeat content, shown in the inner lane (tracks with blue fill) was also high in mini-chromosome contigs, as would be expected, however, several of the other contigs smaller than 2Mb in size had considerable repeat content throughout all or part of their lengths.

A contig of particular interest was BTMP-S13-1 contig 4, shown in the >2Mb plot in the figure and which appears to be a core chromosome but with high mini-chromosome coverage on one end. This is represented in all three mini-chromosome samples shown in the three lanes of the plot, and this region coincides with one of the regions of increased repeat content in that chromosome, as is expected from mini-chromosome sequence. However, the edge of the mini-chromosome-rich sequence region which borders the rest of the contig also has a stretch where the nanopore coverage (green) jumps up to around double that of the surrounding region. This may be indicative of an assembly error, such as a repeat of that segment elsewhere in the genome, perhaps in the mini-chromosome itself. The absence of a significant dip in nanopore coverage between the rest of the contig and the mini-chromosome sequence, however, suggests that it is technically correct that the mini-chromosome sequence is placed on the end of contig 4, and that it is likely biologically correct that mini-chromosome sequence has been integrated into the core genome of contig 4, although probably with some duplication within the mini-chromosome.

The BTGP1-b isolate has many short contigs, and features 48 contigs which are less than 200kb in length, whilst BTMP-S13-1 has only 4 contigs shorter than 200kb. Many of these short contigs in BTGP1-b have high mini-chromosome coverage. None of the contigs larger than 2Mb in length were identified to have significant mini-chromosome coverage. One noticeable contig is contig 20, nearly 800kb in length, which has significant mini-chromosome sequence coverage on both ends, but a dip in coverage in the middle. Nanopore coverage is relatively constant across the length of the contig, whilst repeat content is relatively high.

The BTJP4-1 isolate also displays more small contigs than BTMP-S13-1, with 41 contigs less than 200kb in length for BTJP4-1 compared to the 4 contigs smaller than 200kb in length for BTMP-S13-1. Two of the large contigs, 10 and 19, contain regions on a contig end with greatly increased mini-chromosome coverage. The high-coverage mini-chromosome sequence region which is close to 350kb in length on the end of contig 19, is accompanied by significant repeat content as well as significantly increased nanopore coverage for much of the region, potentially indicative of sequence duplication elsewhere in the genome. There is also a region of noticeably reduced nanopore coverage in the inner border between the high mini-chromosome sequence coverage region and the rest of the contig, which indicates fewer sequences spanning this joining region, and thus displays reduced support for these two distinct regions being part of the same chromosome in the biological reality. This could be a result of an assembly error. The region at the end of contig 10 with increased mini-chromosome sequence coverage is close to 300kb in length, and overlaps a region with increased repeat content, but is not accompanied by increased nanopore coverage except for a significant spike in coverage at the inner border of the region. This spike in coverage at the border region, which reaches approximately double the depth of that in surrounding regions, may indicate duplication of that region in multiple genomic locations. Given the significant repeat content in that region, which is significantly above baseline proportions, it is possible that the segment rich in mini-chromosome sequence appearing on the end of this contig is present due to an assembly error

and does not reflect the biological reality. Despite this, there is no dip in nanopore coverage in the interface between regions which would be a stronger indicator of an assembly error. The proportion of repeat-rich sequence of the region rich in mini-chromosome sequence in this contig is also not significantly above that seen in other regions of the same contig.

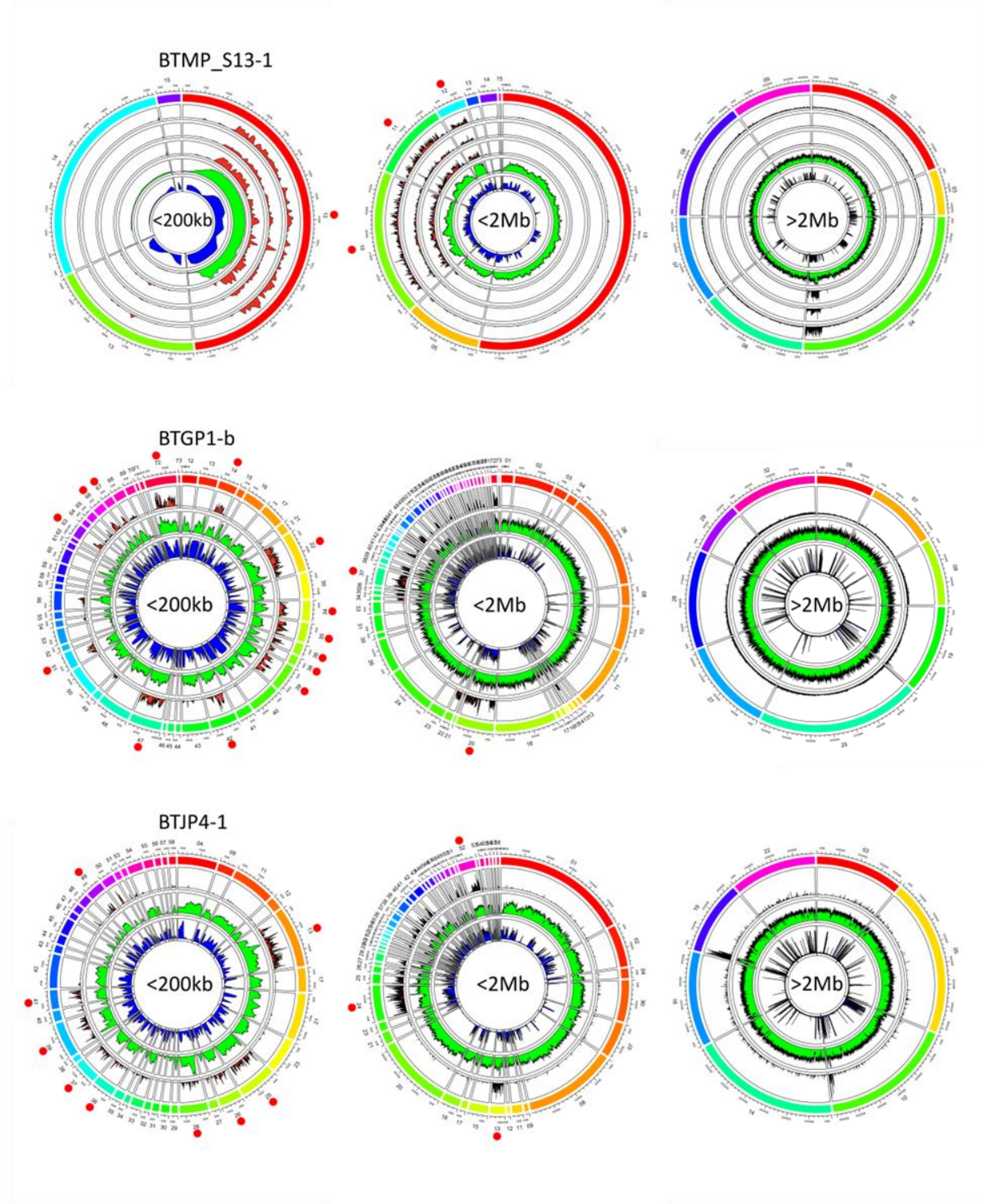

Figure 8. Circos-style plot showing mini-chromosome read coverage corresponding to extracted CHEF gel samples (red outer rings: three tracks in BTMP-S13-1, one track in BTGP1-b and one track in BTJP4-1), nanopore reads (middle green ring) and repeat content (inner blue ring) for the BTMP-S13-1, BTGP1-b and BTJP4-1 Bangladeshi isolates. Contigs smaller than 200kb, smaller than 2Mb and larger than 2Mb are shown in each row from left to right. Contigs smaller than 200kb also appear in the <2Mb plot in addition to the <200kb plot. Mini-chromosome contigs are marked with a red dot, with a table of contigs designated as mini-chromosome contigs given in Appendix 2.

Whilst I did perform alignments between the Zambian isolate genomes and both B71 and Bangladeshi isolates in order to investigate possible Zambian mini-chromosome sequences, I was concerned about the assumptions involved in designating Zambian isolate mini-chromosome contigs on this basis. Due to lacking access to CHEF gel data for the Zambian isolates, I was not able to address this, however, other work (Liu *et al.*, 2023) did subsequently use a similar alignment method as a basis for some of their arguments.

Genomic rearrangements in Bangladeshi isolates were identified

As previously mentioned, and as shown in Figure 9, I identified a significant rearrangement involving the mini-chromosome in the isolate BTMP-S13-1. This is where a sizeable segment from the B71 mini-chromosome aligns to BTMP-S13-1 contig 04, in a region of contig 04 which itself aligns to B71 chromosome 3. A variety of other rearrangements are also visible, such as a large inversion of a segment from B71 chromosome 1 which has been fused to BTMP-S13-1 contig 4, the rest of which aligns to B71 chromosome 3 and the mini-chromosome in B71. B71 chromosomes 2 and 4-7 feature very few structural rearrangements with BTMP-S13-1.

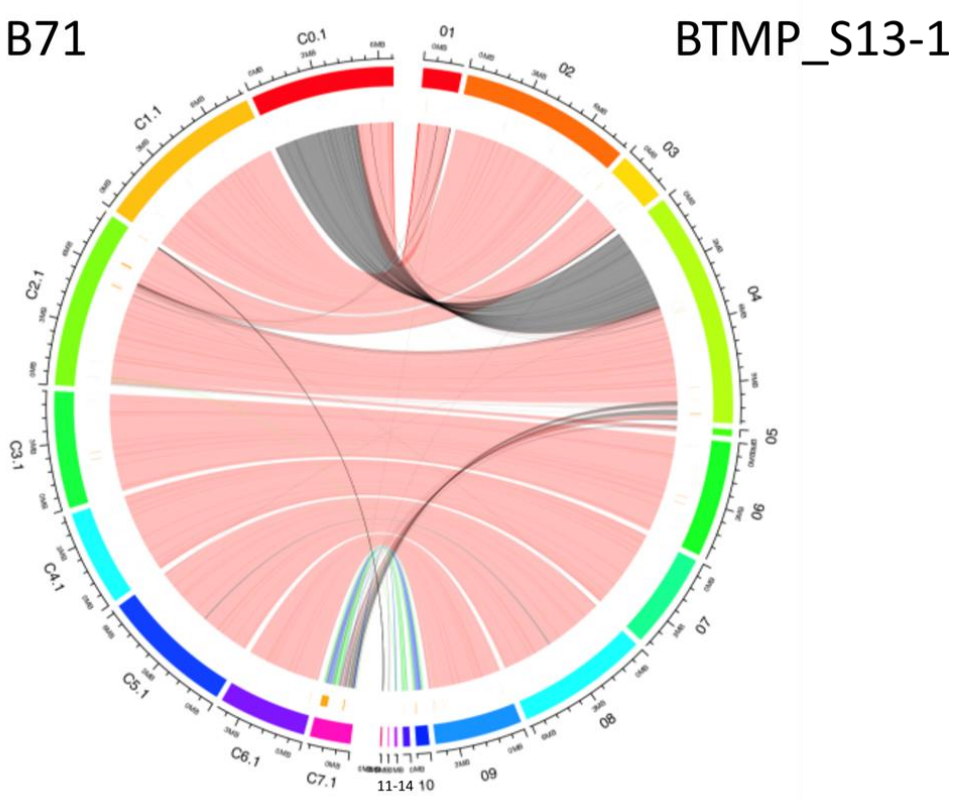

Figure 9. Large-scale genome alignments between BTMP-S13-1 (right) and B71 (left). Ribbons between contigs in each semi-circle show the alignments between contigs from the two genome assemblies, with red (blue if aligning to B71 mini-chromosome) ribbons corresponding to forward alignments, and black (green if aligning to B71 mini-chromosome) ribbons corresponding to reverse alignments.

Given that BTMP-S13-1 was collected in the initial year of the Bangladeshi outbreak, if B71 is representative of the clonal lineage population in South America, this highlights how swiftly significant structural rearrangements between the mini-chromosome and core genome can occur after the outbreak of blast fungus in a new region. However, it is unclear at this time how differentiated the B71 lineage is within South America. This together with the significant likelihood that the Zambian and Bangladeshi populations arose from separate introductions,

along with the genetic differences between these two populations as documented in (Latorre *et al.*, 2023), and the presence of the 12.1.181 isolate in South America, suggests that such genetic differences may have originated prior to introduction into either Bangladesh or Zambia.

Isolate-specific genomic regions were identified

The next stage of analysis was to identify isolate-specific genomic regions via whole genome alignment, when compared to the B71 reference. A minimum contig size of 1kb was used as a filter during the alignment process to reduce noise in the results due to frequent repeat sequences, but this in turn risked introducing many false positive contigs which were shorter than that filtering threshold, usually due to unassembled fragments in the genome assembly.

Figure 10 shows the mean contig number in the B71 lineage isolates available, with a particularly low number for the three Bangladeshi isolates which were used in further analysis marked with red dots. It is also notable that most Zambian isolates have a low contig number. Panel B shows the mean contig length, with those isolates featuring low contig numbers typically also having a longer mean contig length. As shown in panel C, these low contig number isolates usually also displayed lower predicted secretome protein number.

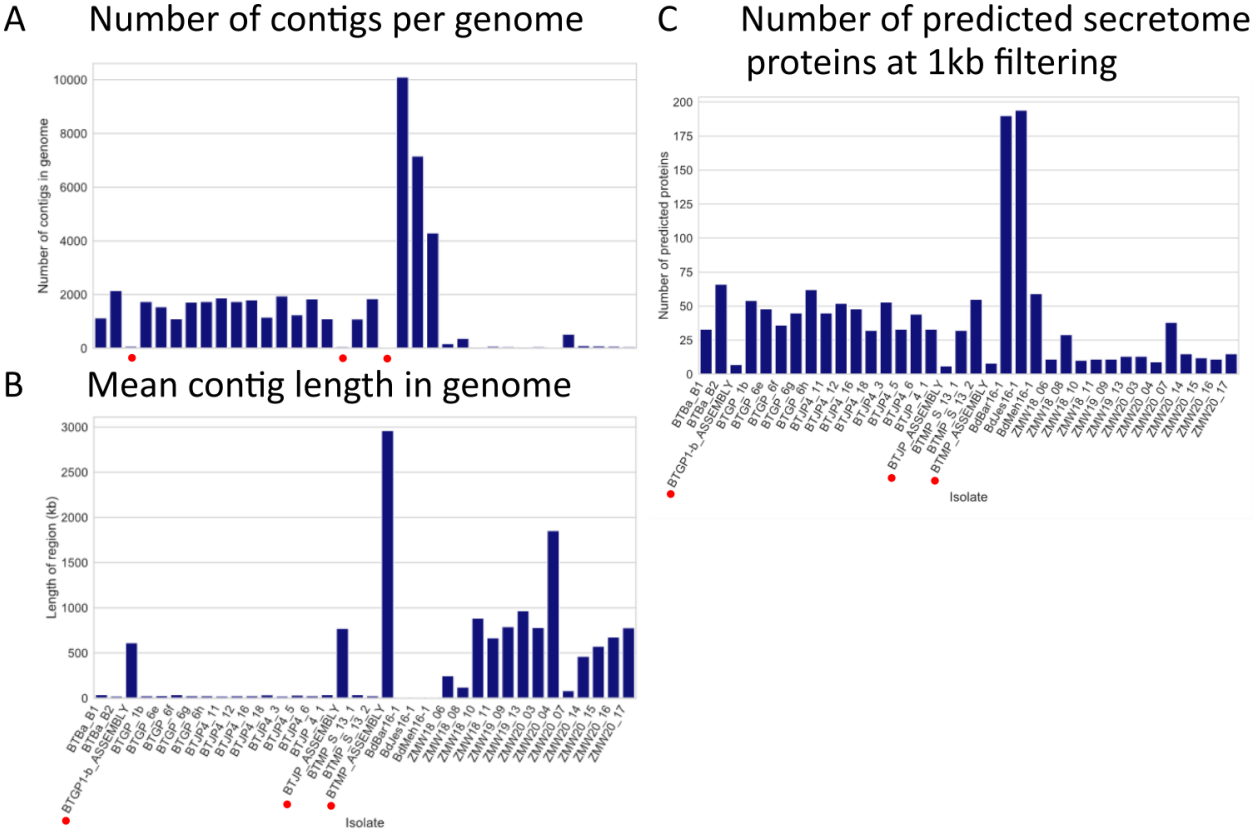

Figure 10. B71 lineage genome characteristics, with red dots marking Bangladeshi genome assemblies used in analysis. (A) Contig number per genome. (B) Mean contig length in each genome. (C) Predicted secretome protein number per genome, at 1kb filtering.

Many of the assemblies available in the dataset consisted of low quality assemblies with many small contigs, which were therefore considered not to be suitable for this analysis. As such, it was necessary to identify only the high-quality assemblies for further work, as shown in Panel

A in Figure 11. As can be seen from this, those isolates with large mean contig size had the fewest secretome (set of secreted proteins) predictions, whilst those with many contigs, and consequently a small mean contig size, also had greatly increased predicted secretome predictions.

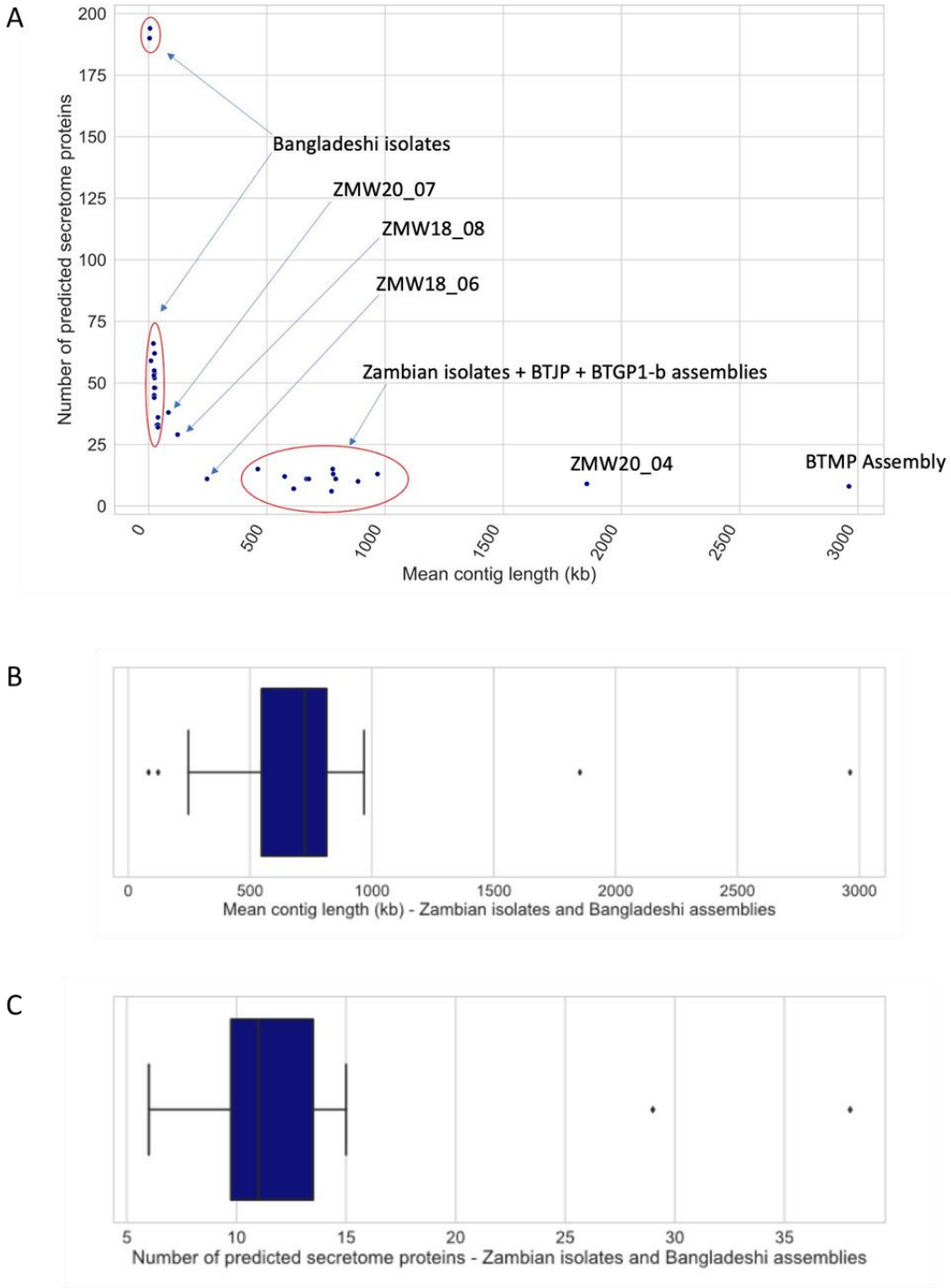

Figure 11. Genomes from the pandemic lineage can be separated into high quality genome assemblies with high mean contig length and few predicted secretome proteins, and a low quality group featuring large numbers of predicted secretome proteins originating as filtering artifacts from small contigs. (A) Mean contig length in each genome shown against the number of predicted secretome proteins, for all initial isolates in dataset. (B) Mean contig length of isolates used for final analysis, with isolates with small fragmented Bangladeshi genome assemblies removed. (C) Number of predicted secretome proteins for isolates in final dataset, with small fragmented Bangladeshi genome assemblies removed.

As a result of these assemblies having a short mean contig length, I discarded the majority of Bangladeshi isolates from the dataset but kept three Bangladeshi assemblies which were of significantly higher quality. Pooling these remaining three Bangladeshi isolates along with the Zambian isolates revealed that ZMW20_07 and ZMW18_08 had a low mean contig length, whilst ZMW20_04 and BTMP-S13-1 had especially high mean contig lengths, which can be seen in Figure 11, panel B. Additionally, as can be seen in Figure 11, panel C, both ZMW20_07 and ZMW18_08 display significantly greater numbers of predicted members of the secretome than the other isolates shown. Consequently, out of the 13 Zambian isolate genomes, I discarded ZMW20_07 and ZMW18_08 from further analysis in order to reduce the risk of false positives in identifying effector candidates.

In an attempt to identify isolate-specific regions from the genomes of the Zambian and Bangladeshi populations, when compared with the B71 reference, I used the Nucmer contig alignment tool from the MUMmer (version 3.1) software to align each isolate to the B71 assembly. The sequences were then filtered to include only those regions in the isolate which did not align to B71, whilst filtering those isolates for 1kb fragments to reduce noise caused by repeat sequences. Consequently, I used this method to identify sequences greater than 1kb in length, and present in the Zambian or Bangladeshi isolate in question, but absent in B71. These sequences were then used as an input for the next stage of the analysis to identify effector candidates.

Population-specific genomic regions contain two effector candidates: Art1_WB_ZM and APiasL3

Through use of the previously-discussed pipeline for unique genomic region identification, I identified 22 effector candidates as being present in either Bangladeshi or Zambian blast fungus genomes but not in B71, and from these, I highlighted two due to their sequence or structural similarity to known relevant proteins, as will be discussed in following chapters. Their presence/absence patterns in the pandemic lineage are shown in panel A in Figure 12, and summarised in panel B. One of these genes, *Art1_WB_ZM*, was contained in Zambian isolates but not in any Bangladeshi isolates or B71. The second, *APiasL3*, was absent in both B71 and the Zambian isolates, but was present in all Bangladeshi isolates. These therefore represent population-specific effector candidates present within genomic segments not shared between the three known populations of the B71 lineage.

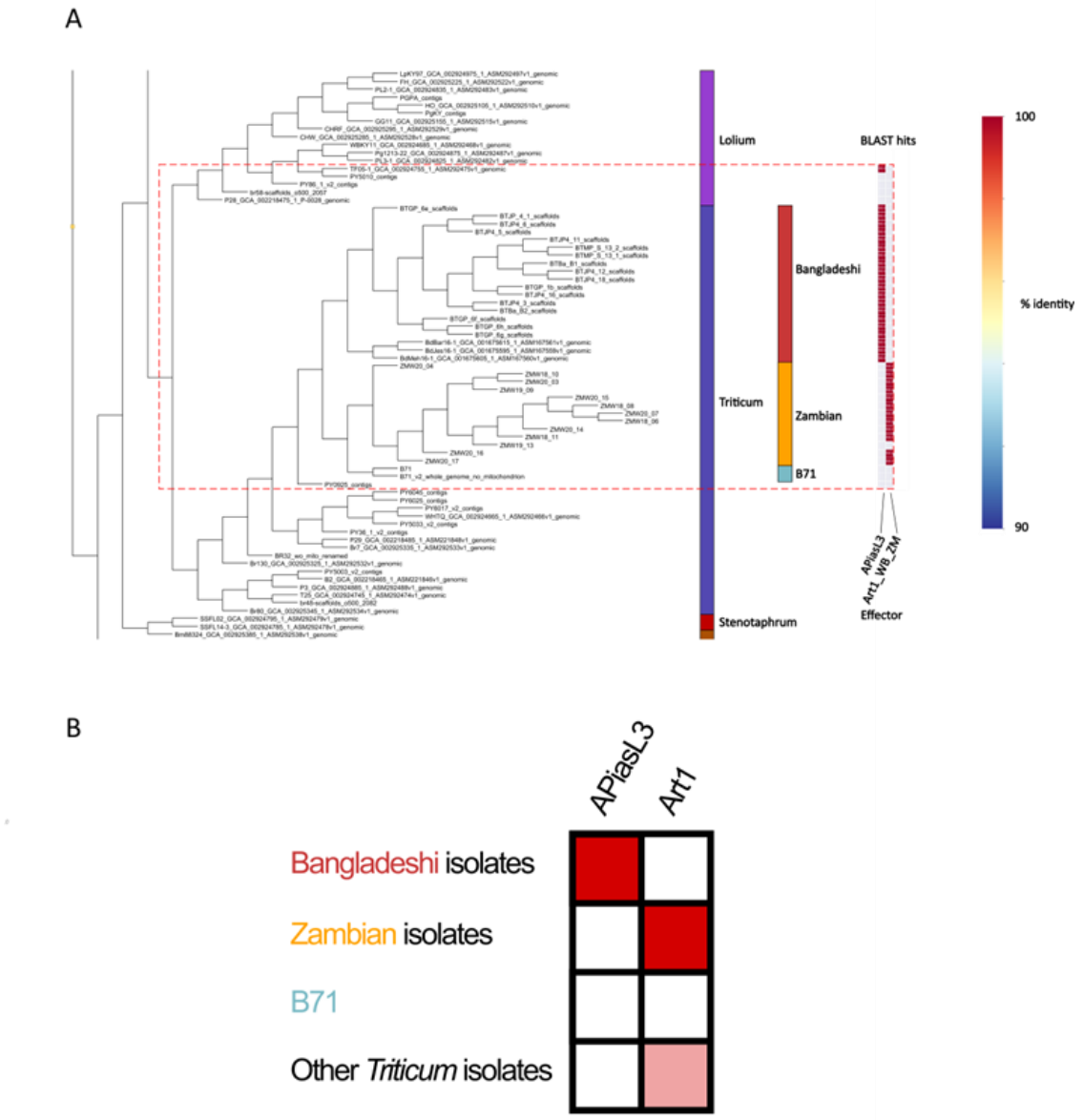

Figure 12. (A) APiasL3 and Art1_WB_ZM distribution throughout pandemic wheat-infecting blast fungus isolates, as determined by BLAST searches. Both effector candidates are absent in B71, but present in either Bangladeshi or Zambian populations. This panel represents a subsection of the blast fungus lineages represented in Figure 13 and Figure 28. (B) Summary schematic of APiasL3 and Art1_WB_ZM presence in Wheat-infecting blast fungus isolates. Dark red implies presence in all isolates, with pink signifying presence in some isolates.

Discussion

Genome rearrangements

The presence of mini-chromosomes within the B71 lineage, but their absence in at least some older isolates such as BR32 and T25 is of interest considering previous studies. As was discussed in chapter 1, previous work (Orbach, 1996) suggested that lack of fertility appears to be linked to the presence of mini-chromosomes. Other work (Gyawali *et al.*, 2023) discussed that there appears to be a period during the 1990s when mini-chromosomes became increasingly prevalent in the wheat-infecting blast fungus population. As discussed in previous work (Valent *et al.*, 2021), the *Triticum* lineage appears to be becoming more aggressive,

though any role the increased presence of mini-chromosomes within the lineage may play in this is unclear.

That the B71 lineage mini-chromosome sequence is not present in certain older wheat-infecting isolates such as BR32 may be indicative that the mini-chromosome did not originate from segmental duplication within early wheat-infecting isolates, but rather from a currently unknown source, such as one which infects on a wild grass species.

The shuffling of genomic material between core genome and mini-chromosomes, such as the apparent inclusion of significant segments of B71 lineage mini-chromosome sequence in the BTMP-S13-1 core genome highlights how early on in the history of an outbreak in a new region significant genomic structural changes can take place. As discussed by (Croll and McDonald, 2012) and as outlined in chapter 1, accessory chromosomes enable structural changes such as segmental duplication and horizontal transfer of genome regions which can facilitate a pathogen's adaptation. It has been recently demonstrated (Barragan *et al.*, 2024) that mini-chromosomes can transfer horizontally from isolates infecting on a wild grass into a clonal lineage infecting on rice, and this may be used by clonal populations to aid adaptation. In addition, mini-chromosomes can potentially influence gene frequency and enable effector gain. These factors can ultimately alter the proteins expressed during infection, potentially allowing clonal lineages to expand their host range, or adapt to counter resistance, both of which are particularly important in an emerging outbreak such as in Zambia and Bangladesh.

Areas for improvement in effector candidate identification

When extracting isolate-specific genomic regions which do not align to the B71 reference, the size of the filtering threshold introduced at the genome alignment stage is critical to avoid either false positive or false negative effector predictions. As was previously discussed, many genomes, primarily from Bangladeshi isolates, needed to be discarded from this initial stage of the analysis due to having a large number of very small contigs. As increasing numbers of high-quality genome assemblies become available, this issue will become less significant.

The method I used to identify effector candidates across blast fungus lineages is highly dependent on BLAST for matching queries with hits across the set of genomes. Further developments of the process which could be beneficial include incorporation of additional tools, such as DIAMOND (Buchfink, Reuter and Drost, 2021) which would significantly increase search speed at a slight sensitivity reduction. Additionally, the pipeline is not currently able to handle introns, and future work should prioritise this to broaden the range of proteins which can analysed. The pipeline presented here does involve a significant amount of manual curation of blast hits in the output, such as to ensure that all residues up to the stop codon have been successfully detected, and removal of the signal peptide. Additional visualisation tools would be beneficial, especially for large sets of query sequences or for many genomes used in the search database.

Effector candidate identification

The presence/absence polymorphism of two effector candidates within the globally-dispersed pandemic lineage could indicate adaptation of different populations of the clonal lineage to their local host and conditions. Altered effector compositions may have developed to evade host R

genes. This variation may be introduced by genomic rearrangements, including the loss or gain of whole regions containing effector sequences, which may involve mini-chromosomes. Additionally, processes such as transposon integration and other genomic rearrangements may induce generation of new effector functions, evasion of immunity or avirulence through inactivation of an avirulence gene, and potentially allowing a pathogen access to new host cultivars or species. As such, a more thorough investigation and characterisation of the specific repeat content and transposons throughout the B71 lineage would be beneficial to understanding the role of rearrangements in facilitating adaptation.

Chapter 4: Investigation of an effector candidate present in Zambian wheat blast isolates

Introduction

As only the second known, sustained expansion onto a new continent from South America for the wheat-infecting lineage, and occurring within a few years of the first such case, the discovery of wheat-infecting blast fungus in Zambia is an interesting opportunity to better understand how this clonal lineage may adapt to the new environments it finds itself in. In the rice-infecting lineage, for example, the three clonal lineages which are geographically dispersed around the world, feature distinctive presence-absence effector polymorphism patterns, which also differ from the ancestral population, believed to aid in adaptation (Latorre *et al.*, 2020). That study also demonstrated that the three clonal lineages carried fewer effector genes than the recombining population did, and suggested this was a result of effector gene loss enabling infection on a previously resistant host, or a polymorphism originating in the originator of the clonal population which was passed to the resulting clonal lineage.

I discussed in chapter 3 how I used a genome alignment-based method to identify genomic regions present in Zambian isolates but absent in the South American B71 isolate. I then predicted effector candidates through translation and signal peptide prediction. The Art1_WB_ZM candidate proved to be of particular interest due to structural similarity between the AlphaFold2 structure I predicted for it, and the structure of HopU1 ART, an ADP-Ribosyltransferase in *Pseudomonas syringae*.

The presence of the Art1_WB_ZM effector candidate in the Zambian wheat blast fungus population, but its absence in the Bangladeshi population and in B71, means it may have developed as a pathogen response to the altered environment and hosts available in Zambia compared with South America. Art1_WB_ZM could be an effector which was present in the South American ancestor, but has been lost in the B71 isolate and the Bangladeshi population, or an effector gained in the Zambian population through inter-lineage transfer of genetic material.

In this chapter, I discuss the prevalence of *Art1_WB_ZM* throughout blast fungus lineages as well as its location in Zambian clonal lineage isolate genomes through the use of BLAST. I then discuss an agroinfiltration experiment to test for toxicity in *N. benthamiana* which did not give any indication of cell death. Following on from this, I discuss my generation of *Art1_WB_ZM* deletion transformants, which were successful, but did not imply any difference between the deletion transformants and wild type in terms of ability to infect on 10 different wheat cultivars. Further, I used AlphaFold 2 to predict the structure of Art1_WB_ZM, although as I discuss in this chapter, the confidence of this prediction is not as high as that for the structure prediction generated by AlphaFold 3, which only became available at the end of this work.

Aims

The aim of the work presented in this chapter is to develop deeper understanding of the Art1_WB_ZM effector candidate through computational and experimental means. I first aimed

to use the pipeline discussed in chapter 3 to map *Art1_WB_ZM* across blast fungus lineages, and then to identify its location in Zambian isolate genomes. I selected agroinfiltration as a method to test for a toxicity role for Art1_WB_ZM without significant prior investment in the experiment. I then wanted to test for any visible virulence changes conferred by the presence of *Art1_WB_ZM*, which I tested through generating transformants where I deleted *Art1_WB_ZM.* I could then perform infection assays, including one featuring 10 different wheat cultivars to compare infection characteristics. The availability of AlphaFold2 also allowed me to predict the structure of Art1_WB_ZM, which was an attractive method to gain insight into structural features without the time and expense required by experimental structure determination, although without the confidence in the results that experimental determination would provide.

Results

Distribution of *Art1_WB_ZM* throughout Blast fungus lineages

As Figure 13 shows, I identified that *Art1_WB_ZM* is present throughout much of the *Triticum* lineage, although it is notably absent in both B71 and the Bangladeshi population. I also found it is present amongst 13 *Oryza* lineage isolates including several samples originating in Italy (work involving these samples is further described in (Barragan *et al.*, 2024)), and in addition, examples of it can be identified within 2 *Brachiaria* and 5 *Setaria* isolates, albeit containing polymorphisms. Three isolates belonging to the more distantly-related *Digitaria* lineage also contained *Art1_WB_ZM*.

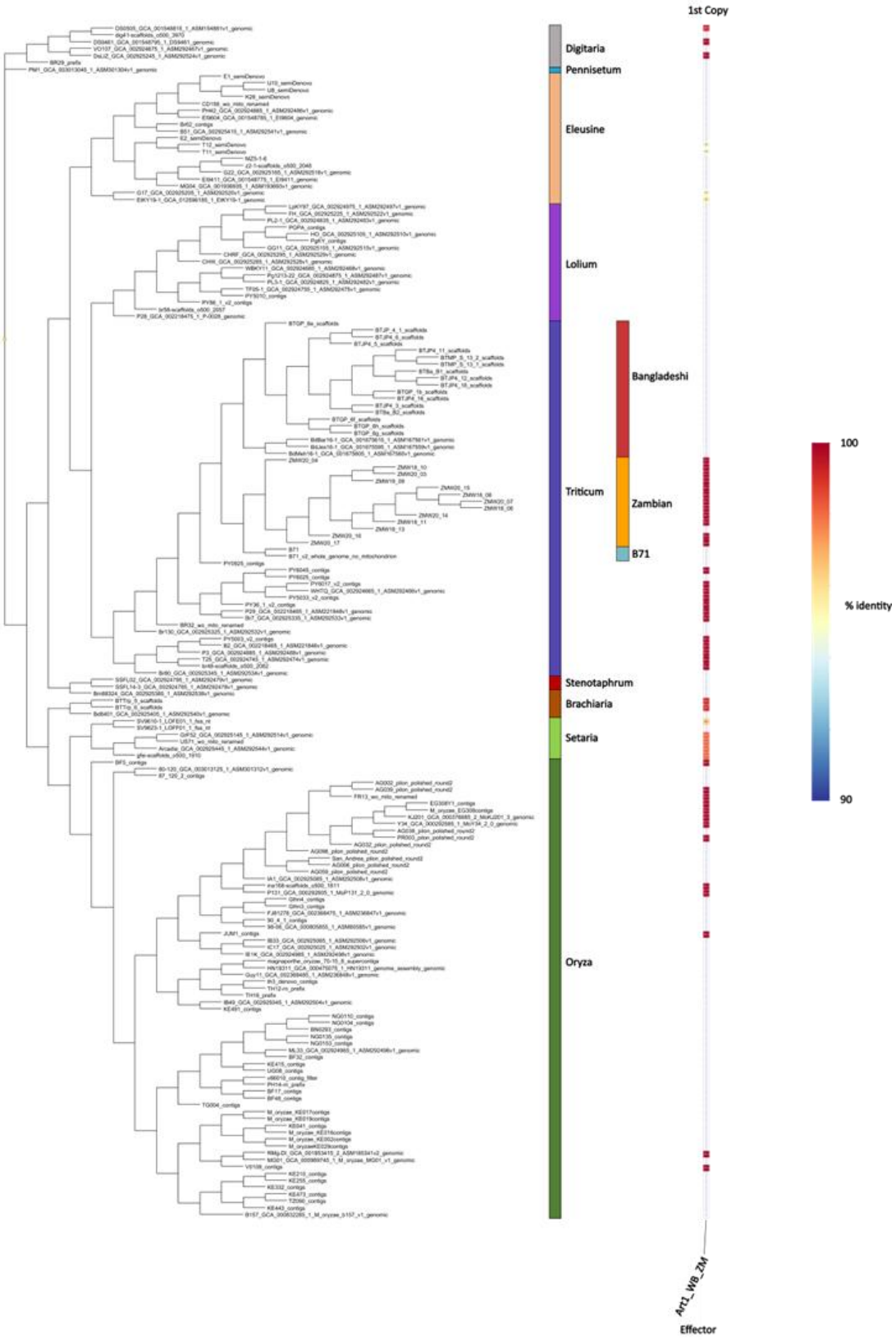

Figure 13. Presence/absence patterns of Art1_WB_ZM throughout blast fungus host-specific lineages, with a notable presence in the Zambian wheat-infecting population in the B71 lineage. There is no presence in B71 or the Bangladeshi sub-lineage of wheat-infecting sub-lineage, but there is presence within isolates from other host-specific lineages, such as the Oryza lineage.

My initial analysis used BLAST to identify presence/absence of the effector candidates through the pandemic clonal lineage, and identified *Art1_WB_ZM* as present in all analysed members of the Zambian population except for ZMW19_13. Following this, I mapped short read data, revealing that ZMW19_13 did contain *Art1_WB_ZM*. The BLAST-based analysis returned a match to *Art1_WB_ZM* with 100% coverage (the proportion of the query which aligns with the match) but only 88.7% identity (the percentage of positions which are identical between the query and match sequences). This means that the entire sequence present in ZMW19_13 aligns with *Art1_WB_ZM*, but 88.7% of the sequence is polymorphic. Due to this being less than the 90% filtering threshold which I had previously selected as an appropriate cutoff value to assign presence/absence, this match was marked as absent in the BLAST-derived presence/absence heatmap (Figure 13).

I used the available ZMW19_13 short read sequences and mapped them against *Art1_WB_ZM* to test if the reduced percent identity in that isolate which was detected by BLAST, was due to an assembly error, or if it reflects a polymorphism between that isolate and other Zambian isolates. This was because any error introduced during the genome assembly process would not be observable in the short read sequences. When I mapped ZMW19_13 short read sequences against *Art1_WB_ZM*, there was no observable coverage drop over the region. There were also similar levels to the short read sequences from three other Zambian isolates which I used as controls, but reads mapped from a Bangladeshi isolate to *Art1_WB_ZM* revealed no mapped reads. SNPs were not noticeable in the short read mapping for ZMW19_13. This pattern matches the anticipated results if *Art1_WB_ZM* is present in ZMW19_13, and I therefore concluded that *Art1_WB_ZM* is present in all Zambian isolates analysed.

Art1_WB_ZM is located on a contig aligning to B71 chromosome 2 with a Zambian-specific end region

I conducted BLAST searches using the *Art1_WB_ZM* sequence as a query in order to identify the location of the effector candidate within Zambian isolate assemblies. As can be seen in Figure 14, *Art1_WB_ZM* is present in a genomic region at the start of a large contig in isolates ZMW19_09 and ZMW20_04. Whilst the majority of the contig is aligned with core chromosome 2 in B71, and is conserved amongst Bangladeshi isolates like BTMP-S13-1, the genomic segment directly containing *Art1_WB_ZM* is absent in both B71 (represented by a shaded blue bar over Zambian contig coordinates) and in the Bangladeshi isolates.

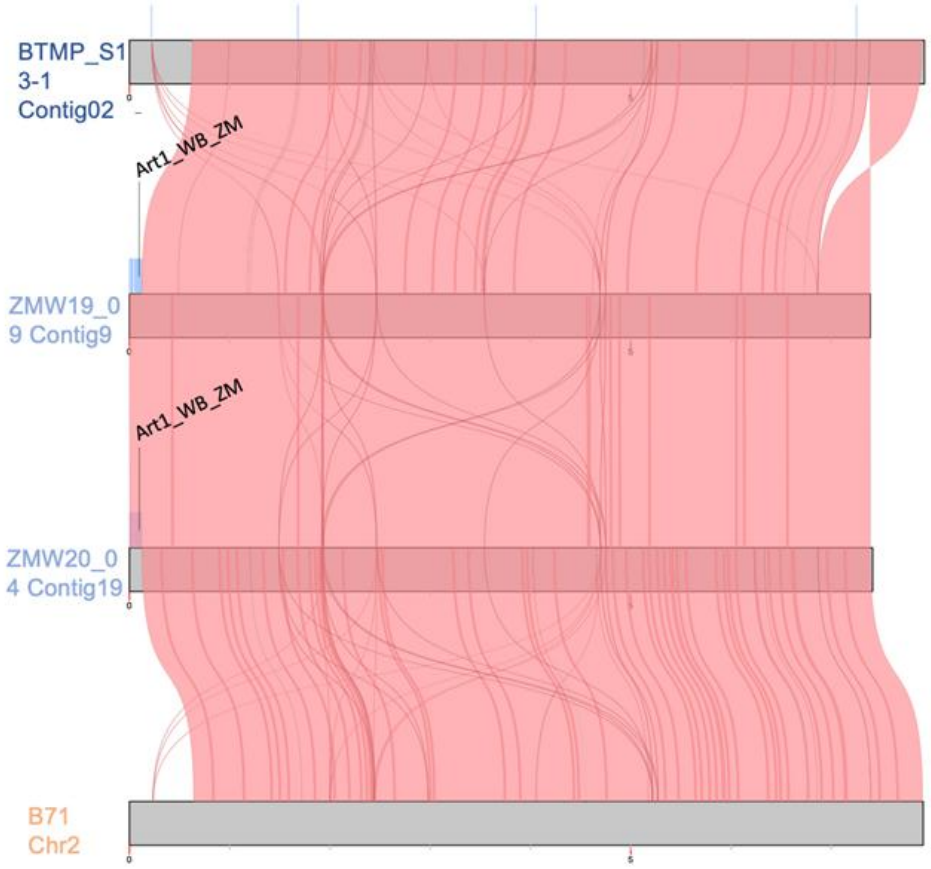

Figure 14. Location of Art1_WB_ZM in two Zambian isolates, shown with pairwise contig alignments (red bands between contigs) to B71 and BTMP-S13-1. Blue bars over contigs highlight regions which don't align to B71. Of particular note is the initial region of contig9 in ZMW19_09 and Conting19 of ZMW20_04 that contains Art1_WB_ZM, each of which aligns to the other, but are absent in both B71 and Bangladeshi isolates.

An agroinfiltration experiment did not detect cell death due to Art1_WB_ZM

Due to the predicted structural similarity of Art1_WB_ZM to a known member of a family of bacterial toxins operating by modifying core host proteins, I wanted to test for toxicity induced by Art1_WB_ZM when expressed in plant cells, and agroinfiltration was a suitable method to quickly test this hypothesis. Agroinfiltration is an experimental technique used to induce transient gene expression in plant tissues, through transferring genetic material via *Agrobacterium tumefaciens*. Consequently it can be used to test the effects of a protein encoded within a plasmid carried by the bacterium when expressed within plant cells.

The results of this experiment, shown in Table 1, were that the positive control leaf infiltration sites displayed cell death, and the negative control sites did not, as expected. Both Art1_WB_ZM inoculation sites did not display any sign of cell death.

Table 1. Results from agroinfiltration experiment with pAM8 Art1_WB_ZM-containing plasmids, and the positive control AVR-PikD with Pikm, and the negative control AVR-PikC with Pikm. No variance in result was noted for any leaves. A '+' indicates a cell death response, whilst '-' indicates no response. The two biological replicates of pAM8 were infiltrated in two spots on each leaf.

Infiltration spot	AVR-PikD + Pikm (positive control)	AVR-PikC + Pikm (negative control)	pAM8 replicate 1	pAM8 replicate 2
Cell death response	+	-	-	-

Transformants were successfully generated for *Art1_WB_ZM*-deletion in one Zambian isolate

My initial attempt at generating *Art1_WB_ZM*-deletion transformants from the ZMW18_10 wild-type isolate produced nine candidates which were able to grow on hygromycin-containing medium, although attempted verification using PCR with primers corresponding to a diagnostic sequence present only in successful transformants proved inconclusive.

In a second attempt to generate *Art1_WB_ZM*-deletion transformants, I was successful in generating a pool of 75 transformants able to grow on Hygromycin-containing medium out of a total of 90 potential transformants. Out of those, ten (Zt5, Zt10, Zt12, Zt18, Zt19, Zt20, Zt22, Zt23, Zt24, Zt28) were indicated to have undergone effector replacement by the Hygromycin resistance gene through PCR tests with diagnostic primers, and three promising candidates were selected for further analysis.

The success of transformant generation was tested with Southern blotting and sequencing

In order to test the success of transformation, PCR was performed with two sets of primers corresponding to diagnostic sequences overlapping and flanking the transformed region, such that only a successful transformant should display a band in that lane. These correspond to lanes d1d2 (lane 2 – upstream flank, UF diagnostic) at 1.35kb and d3d4 (lane 3 – downstream flank, DF diagnostic) at 1.2kb in each transformants' position, as seen in Figure 15. The first lane corresponds to a sequence present only in the mini-chromosome (primers contributed by Thorsten Langner) serving as a control region which should always be present in both the wild type and transformants derived from it. In the figure, ten isolates (highlighted in red boxes) containing bands in the expected locations were shortlisted.

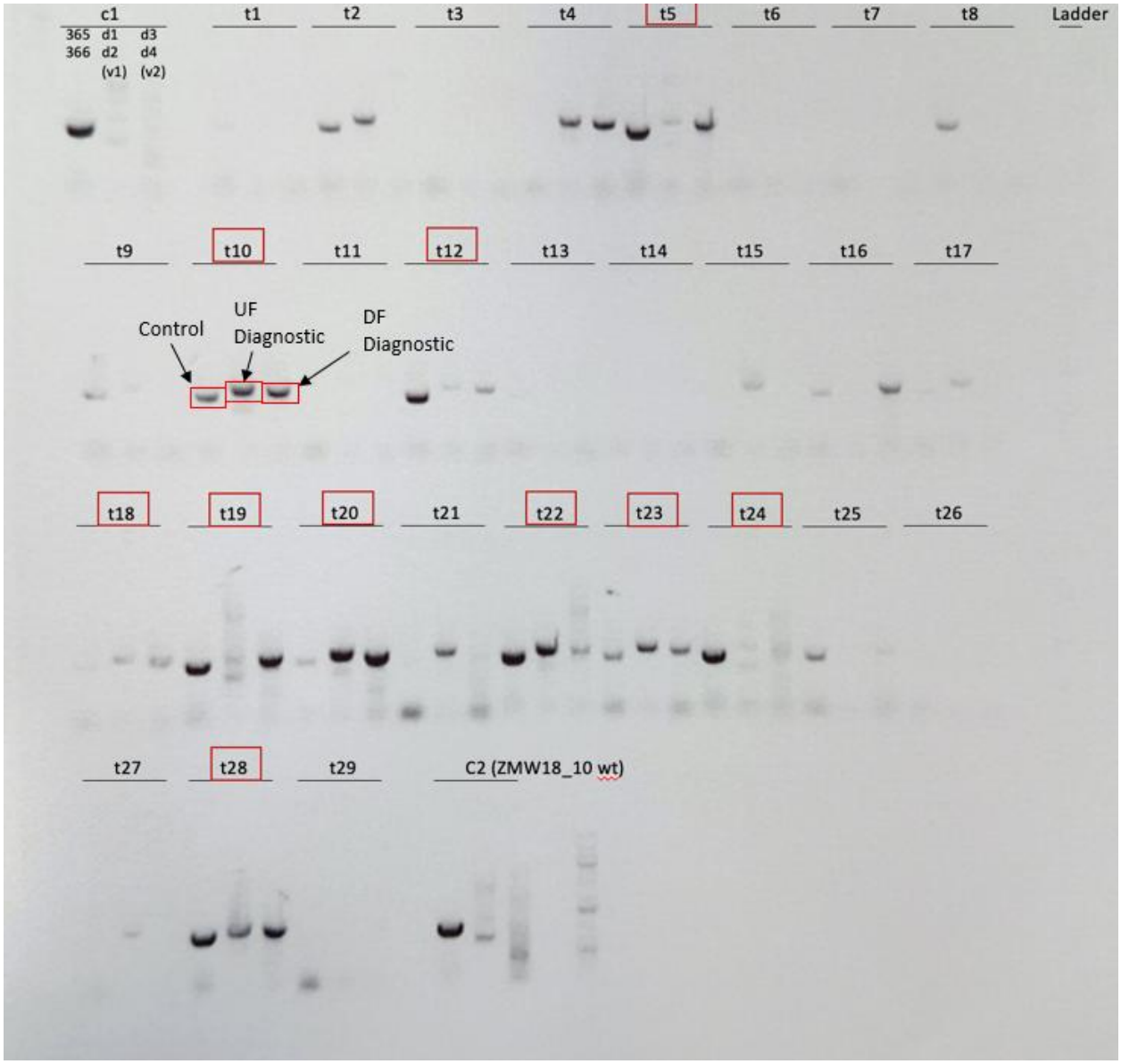

Figure 15. Diagnostic PCR gel displaying a control lane, upper flank and down flank diagnostic lanes, for each of the 29 transformants and 2 controls (C1 and C2). Each of the three lanes for each isolate should contain a band, as shown in the t10 sample, if the transformation proceeded correctly.

I selected three of these transformants (t10, t20 and t28) for further analysis, which were designated as Zt10, Zt20 and Zt28 respectively in further work. I extracted the d1d2 and d3d4 gel bands from the gel and measured the DNA concentration to be between 12-36ng/µl. Whole genome DNA from each of the three shortlisted transformants was extracted using CTAB.

The diagnostic region was also isolated from the genomic DNA of each candidate transformant and sequenced to ensure there were no gaps or unexpected artifacts arising from the transformation.

A strategy was developed to use Southern Blotting to test the success of transformant generation, as described in the methods chapter. Southern blotting is a technique allowing visualisation of a certain sequence of DNA within a sample, such as to determine gene presence. This is done through DNA fragmentation of the DNA sample using restriction enzymes, then gel electrophoresis to separate those fragments by length. Next, for

visualisation the fragments in the gel are transferred across to a filter paper or polymer sheet, and then the DNA sequence of interest can be visualised using a labelled complementary strand, such that a band in the final image will only be observable if that sequence was present in the sample. In this case, the method did not yield positive or negative results due to a failure of the final gel to run properly, and the resultant image shown in Figure 16 is consistent with non-specific probe binding. Ultimately a switch of method to use genome sequencing was favoured instead.

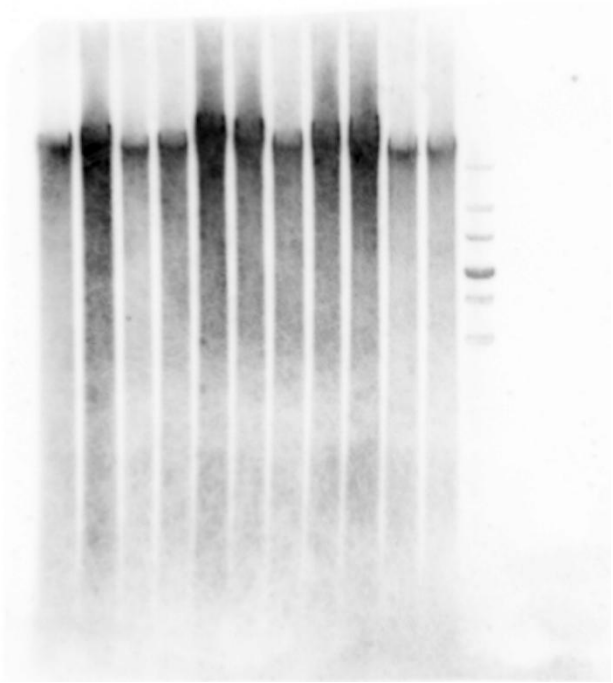

Figure 16. Due to a lack of specific labelled complementary strand bands appearing, the output from Southern blotting did not yield positive or negative results. This is consistent with non-specific probe binding. Subsequent approaches focussed on a sequencing-based method instead.

I used a Nanodrop machine to test DNA concentration and A260/280 values. Each of the three samples had a measured concentration greater than 290ng/µl, but their 260/280 ratios were between 0.12 and 0.26. The 260/280 ratio measures the ratio of absorbance at 260nm and 280nm, and is used to measure purity in DNA samples, with values over 1.8 being considered suitable for sequencing due to lower amounts of contaminants. These obtained values are not sufficient for sequencing, and I therefore used the more-suitable QIAGEN DNeasy Plant Mini Kit to extract DNA from the transformants using liquid Nitrogen to grind the samples, eluting into water. On this second attempt I obtained sufficient DNA concentration (247.7ng/µl) and purity (260/280 ratio = 1.93) for sequencing only for transformant Zt10. I repeated the DNA extraction again, and on this attempt obtained 260/280 ratios of 1.8 for both Zt20 and Zt28, with the DNA concentration in both cases being above 25ng/µl. I sent the samples (final DNA concentration and purity values are shown in Table 2) to Novogene Co., Ltd for whole genome sequencing using Illumina.

Table 2. DNA concentration and purity values used for whole genome sequencing.

Sample	Concentration (ng/microlitre)	260/280	260/230
Zt10 (v2)	247.7	1.93	1.01
Zt20 (v3)	36.32	1.80	1.07
Zt28 (v3)	26.92	1.80	1.10

I mapped whole genome sequencing data from Illumina to both the wild type assemblies and a simulated transformant sequence where the expected Hygromycin sequence replaced the original effector sequence, as shown in Figure 17. This demonstrated that for the transformants, there was no coverage over the *Art1_WB_ZM* sequence but there was an approximately constant coverage when mapping to the simulated transformant sequence containing the Hygromycin gene. Meanwhile, there was uniform coverage across the *Art1_WB_ZM* locus for the wild type control sequence, but a lack of coverage for the simulated transformant sequence.

An artifact visible in Figure 17 is the coloured vertical bars shown in the rows corresponding to mapping to the simulated transformant genome, which signify SNPs. This is a result of plotting the data using the wild type genome as the basis for the plotting coordinate system in the Integrative Genomics Viewer (IGV) version 2.17.0 (Robinson *et al.*, 2011) system, rather than the simulated transformant, and if that is used instead the apparent SNPs transfer to the mapping to the wild type rows. It is not reflective of any real SNPs in this data. This provides evidence for the successful replacement of *Art1_WB_ZM* by the Hygromycin gene in the three transformants analysed.

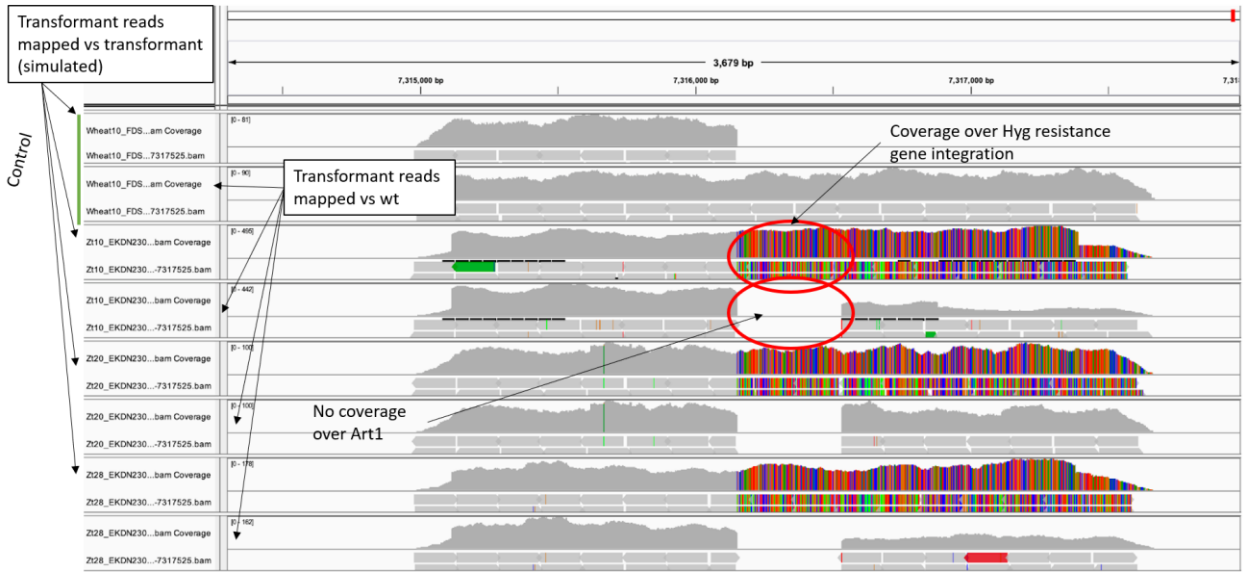

Figure 17. Mapping of Art1_WB_ZM-deletion transformant short read sequencing data onto both wild type and simulated-transformant ZMW18_10 genomes.In each case, the mapping to the wild-type genome displays a distinctive gap over the effector sequence, whilst coverage is continuous when mapped to the simulated-transformant sequence. Image created by author using IGV (version 2.17.0), and annotated for clarity.

Infection assays with *Art1_WB_ZM* -deletion transformants did not indicate significant differences in infection vs wild type

Using two wheat cultivars (Fielder and Chinese Spring) and two barley cultivars (Golden Promise and Nigrate), I conducted a leaf-drop infection assay, shown in Figure 18, as a preliminary experiment to test if absence of the *Art1_WB_ZM* effector candidate alters pathogen virulence when compared with the wild-type ZMW18_10 isolate carrying *Art1_WB_ZM*. Three transformants (Zt10, Zt20 and Zt28) were used and appear to display the same infection characteristics as the wild-type. On each successive leaf on each plate, droplet position was shifted downwards by one place, with the bottom-most droplet moving to the top of the leaf, which was done to control for drop position on the leaf, as shown in panel A of Figure 18. BTMP-S13-1, serving as the positive control, infected as expected on Fielder (wheat), Chinese Spring (wheat) and Nigrate (barley), however, on Golden Promise (barley), it did not infect. As expected, the drop without *M. oryzae* spores (negative control - marked as water in the figure) did not result in infection symptoms. Therefore, there is no evidence to suggest that effector deletion resulted in any alteration to virulence on these cultivars.

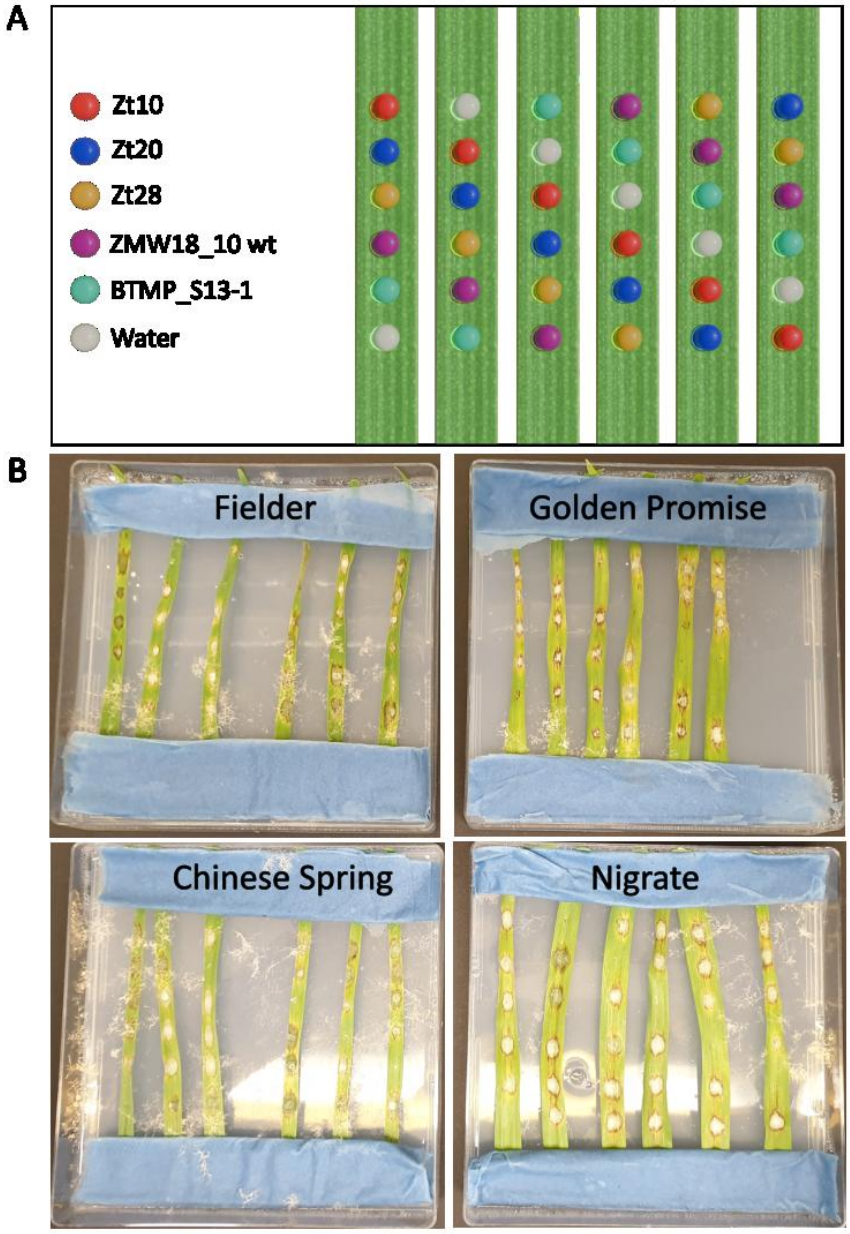

Figure 18. Panel A illustrates the setup of the infection assay using two wheat cultivars (Chinese Spring and Fielder) and two barley cultivars (Nigrate and Golden Promise). This used ZMW18_10 (wild type), three Art1_WB_ZM deletion transformants of the same isolate, a water control and a Bangladeshi isolate (BTMP-S13-1) as a further control isolate. Images were taken 4dpi. On each plate, each leaf from left to right has had the drops displaced one position downwards, with the bottom-most drop being moved to the top, which is illustrated in the top legend (produced using the Blender 4.0 3D modelling software). Panel B shows the resulting lesions after performing the experiment

Table 3. Results from initial infection assay. Each '+' indicates one resistance spot on a leaf, and each '-' indicates an absence of a visual response.

		Cultivar			
Isolate		**Fielder**	**Chinese Spring**	**Golden Promise**	**Nigrate**
	Zt10	++++++	++++++	++++++	++++++
	Zt20	++++++	++++++	++++++	++++++
	Zt28	++++++	++++++	++++++	++++++
	ZMW18_10 (wild type)	++++++	++++++	++++++	++++++
	BTMP_S13_1 (Control)	++++++	++++++	- - + - - -	++++++
	Water (Control)	- - - - - -	- - - - - -	- - - - - -	- - - - - -

I conducted a follow-up experiment using ten wheat cultivars and two of the three shortlisted *Art1_WB_ZM*-deletion transformants, in an attempt to determine if a similar experiment using a wider array of hosts could detect any differences in ability to infect. The results found both transformants infecting to the same degree as for wild type ZMW18_10 (and BTMP-S13-1 as a control), with all controls outputting expected results. Size differences in lesions were not investigated.

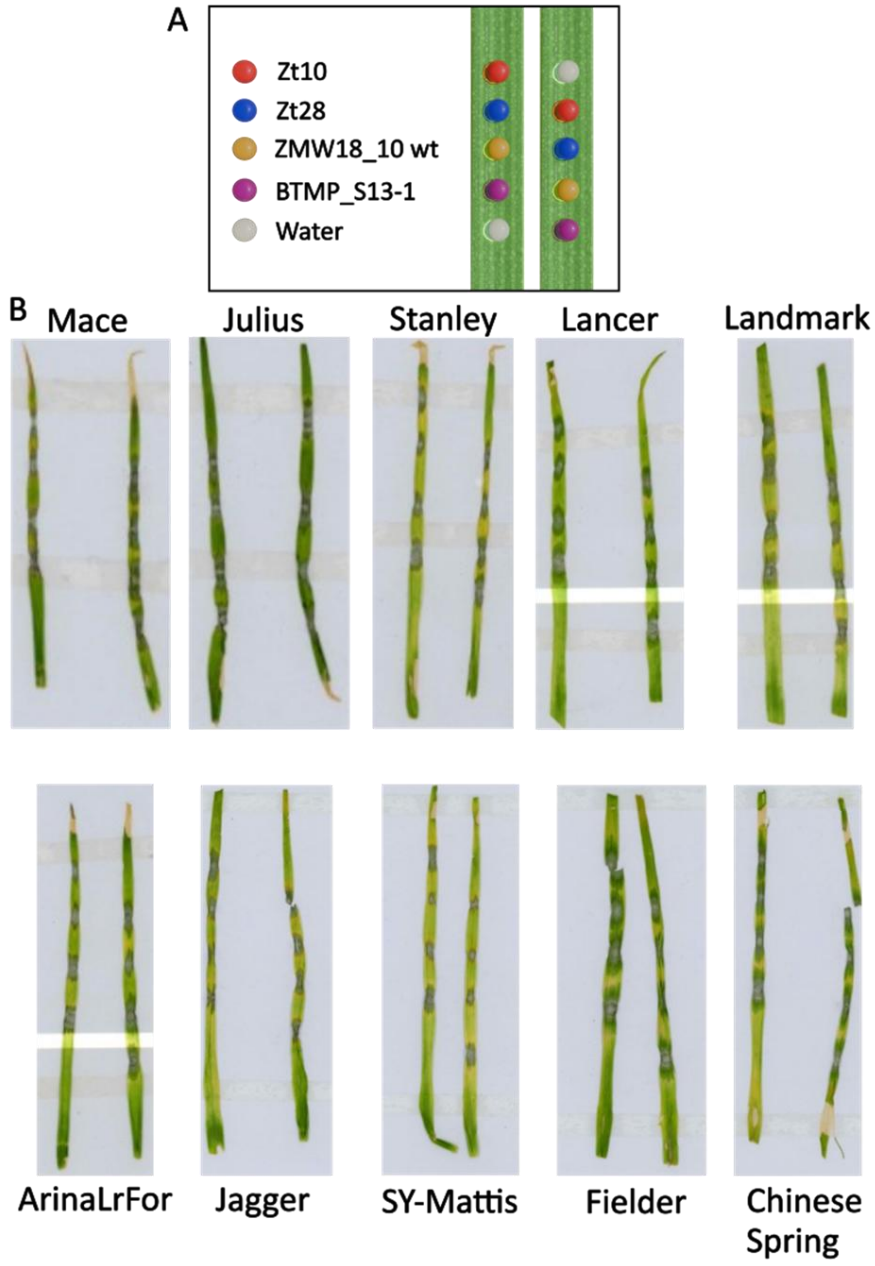

Figure 19. Infection assay results for 10-wheat cultivar test. Panel A illustrates the experimental setup, with two leaves used for each cultivar and 5 droplets used for each leaf. Droplets were shifted by one position between the first and second leaf in each pair. The droplet position illustration was produced using the Blender 4.0 3D modelling software. Panel B shows the experimental results, divided by cultivar.

The experiment indicated the *Art1_WB_ZM* deletion transformants were able to infect to a similar degree as the wild-type isolate which the transformants were generated from. No significant differences between cultivars were detected, and the positive and negative controls behaved as expected.

Table 4. Results from infection assay. Each '+' indicates one resistance spot on a leaf, and each '-' indicates an absence of a visual response.

		Cultivar									
Isolate		**Mace**	**Julius**	**Stanley**	**Lancer**	**Landmark**	**ArinaLrFr**	**Jagger**	**Sy-Mattis**	**Fielder**	**Chinese Spring**
	Zt10	++	++	++	++	++	++	++	++	++	++
	Zt28	++	++	++	++	++	++	++	++	++	++
	ZMW18_10 (wild type)	++	++	++	++	++	++	++	++	++	++
	BTMP_S13_1 (Control)	++	++	++	++	++	++	++	++	++	++
	Water (Control)	--	--	--	--	--	--	--	--	--	--

An additional infection assay was conducted, which used four cultivars, selecting five leaves for each, and testing two Zambian isolates (ZMW20_14 wild type and ZMW18_10 wild type) and two Bangladeshi isolates (BTJP4-1 wild type and BTMP-S13-1 wild type), with the Chinese Spring, Fielder, Nigrate and Golden Promise wheat and barley cultivars, as used previously. The original intention of this experiment was to test what I had believed at the time was a natural knock-out of *Art1_WB_ZM* within the ZMW19_13 isolate, and compare this with wild type Zambian isolates which did carry *Art1_WB_ZM*. However, the ZMW19_13 isolate could not be used in the experiment as planned, since it sporulated four days later than anticipated. I subsequently also realised that ZMW19_13 did carry *Art1_WB_ZM*, so the rationale of the experiment was not valid. As such, the experiment carries no value in terms of its original intention, but the additional isolate ZMW20_14 was not included in the other infection assays.

The infected leaves (five used per cultivar) did not show any observable variance between any of the isolates, and all caused lesions consistent with infection, as expected, whilst the water control did not cause lesions. The exception to this is the BTMP-S13-1 isolate, which did not cause infection symptoms on some of the Fielder wheat leaves, for unknown reasons, although I did not observe this lack of infection in other infection assays involving this isolate and cultivar combination. There was no observable difference between wheat and barley cultivars, and consequently it can be observed that the ZMW20_14 isolate infects on Fielder and Chinese Spring wheat similarly to ZMW18_10. The final analysis was performed four days post infection (dpi).

Computational prediction of the Art1_WB_ZM structure and screening for similar structures indicated structural similarity with HopU1 ART

I predicted the structure of the Art1_WB_ZM sequence (with signal peptide removed) using ColabFold-AF2 (Mirdita *et al.*, 2022), a Google Colab notebook running AlphaFold2, optimised for ease of use (Figure 20). Using this predicted structure, Michelle Hulin identified a similarity with the structure of the HopU1 ART (ADP-Ribosyltransferase) domain found in *Pseudomonas syringae*, using the software HHPred. It is notable, however, that the active residues found in the recognised forms of the ART family are not present in Art1_ZM_WB. Whilst highly sequence divergent, the ART superfamily has a core fold which is conserved (Aravind *et al.*, 2015). HopU1 ART targets RNA-binding proteins and for full virulence on *Arabidopsis thaliana*, the pathogen requires it (Nicaise *et al.*, 2013). Figure 21, which was obtained from (Jeong *et al.*, 2011), shows the HopU1 structure. The C terminus region, shown in red, is the region displaying structural similarity to the AlphaFold2 predicted Art1_WB_ZM structure.

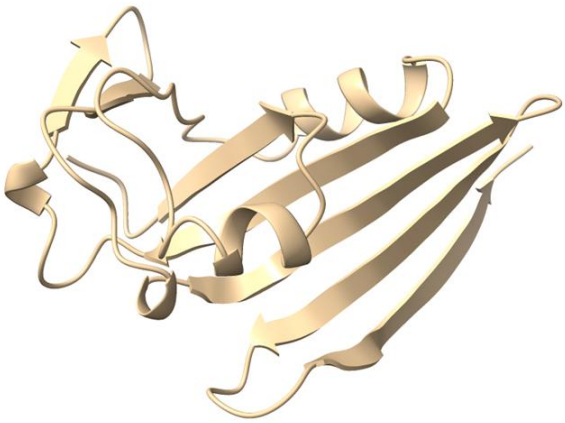

Figure 20. Structural prediction on Art1_WB_ZM using ColabFold-AF2.

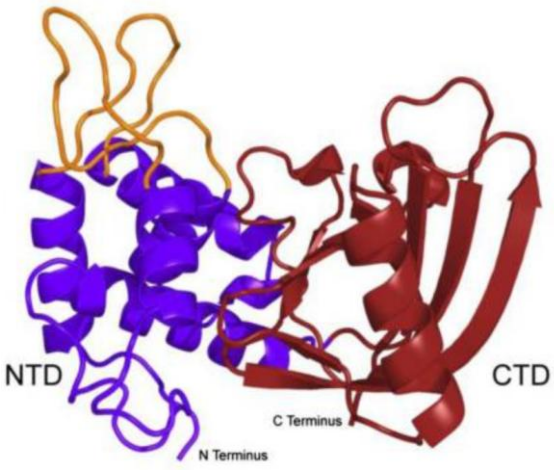

Figure 21. HopU1 structure, obtained from figure 1 in (Jeong et al.*, 2011). Their work shows the C terminus in red, which is the domain showing structural similarity to Art1_WB_ZM.*

Figure 22 illustrates the same structure as Figure 20 (predicted by ColabFold-AF2), but in the latter case the structure is coloured by pLDDT value. As can be seen from this, several regions of the prediction are shown in blue, corresponding to a good prediction confidence with pLDDT values between 70 and 90, although other regions are significantly less confident at between 50 and 70. Regions with values less than 50 are very low confidence.

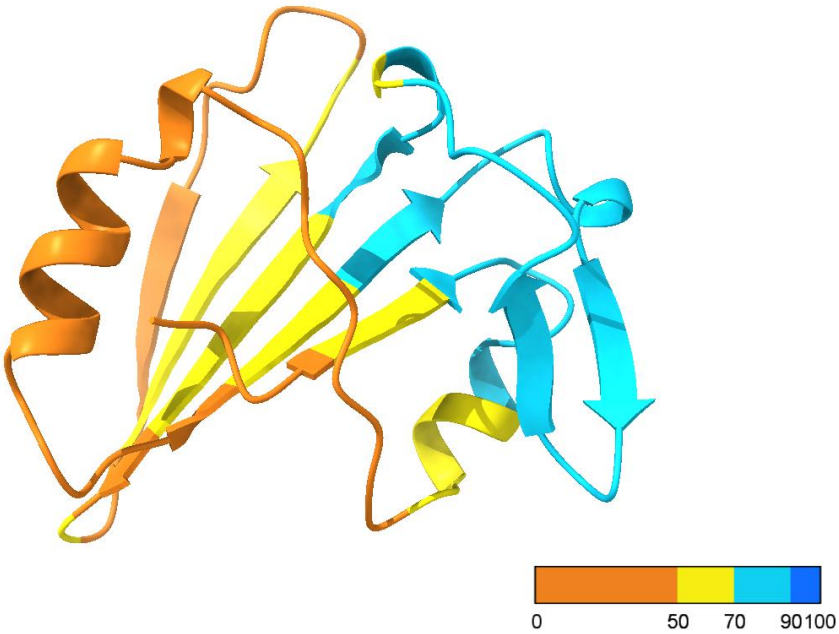

Figure 22. Original structural prediction on Art1_WB_ZM using ColabFold-AF2, coloured by pLDDT value. Visualisation produced using ChimeraX 1.5.

I subsequently used AlphaFold3 when it became available to predict the structure of Art1_WB_ZM, with the result shown in Figure 23 (superimposed over the AlphaFold2 model which is in grey and is semi-transparent). This is also coloured by pLDDT value, and it is of a significantly higher confidence than the previously predicted version of the structure, with most pLDDT values over 90. In the case of this second structure, the pTM (predicted template model) value, which is a measure of the whole structure accuracy, is 0.77. Due to being greater than 0.5, it is likely that this AlphaFold3 prediction is significantly closer to the true structure than the original structure generated using ColabFold-AF2. Panel B in Figure 23 shows the predicted aligned error (PAE) matrix for the AlphaFold3 structure prediction, where lower values correspond to increased confidence in relative positioning between residues across the structure. For this structure, but not for the AlphaFold2 prediction, the PAE values were low across most of the structure.

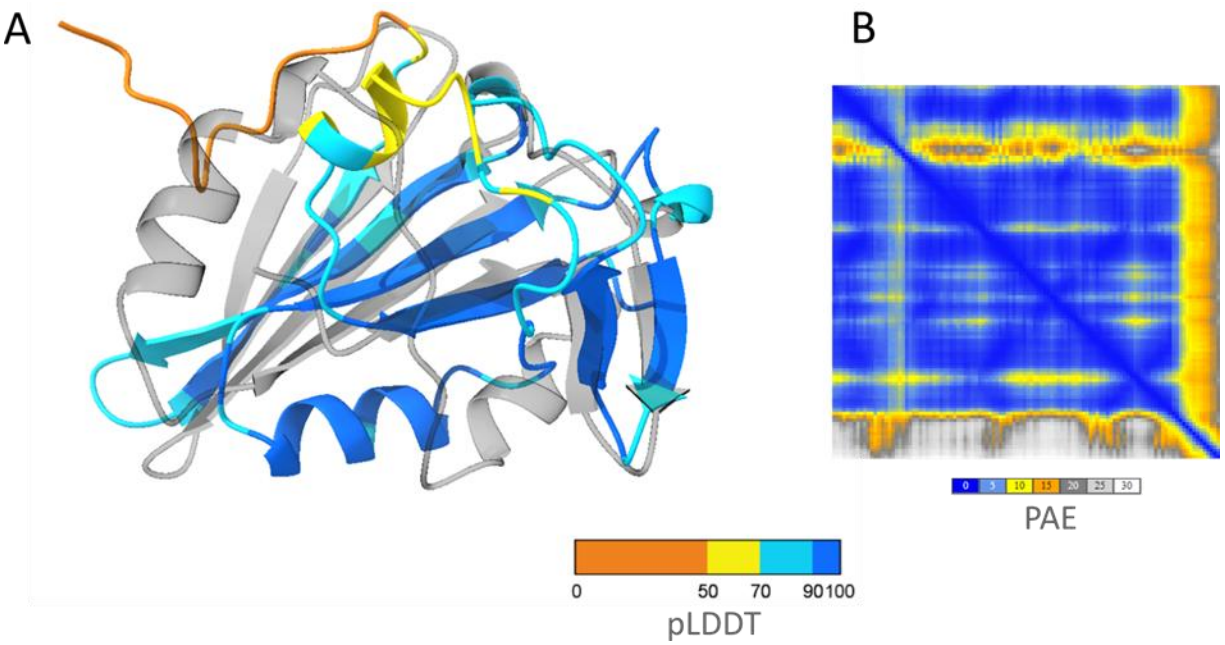

Figure 23. (A) Structural prediction on Art1_WB_ZM using AlphaFold3, coloured by pLDDT value and displaying significantly greater prediction confidence than the AlphaFold2 structure. The AlphaFold2 structure is shown in grey and semi-transparent. (B) The PAE values corresponding to the AlphaFold3 structure prediction shown in panel A. The values for most of the structure are very low, representing increased confidence in the relative positions of residues in different locations in the structure.

Discussion

Art1_WB_ZM presence across blast fungus lineages and genomic location

The absence of *Art1_WB_ZM* in the Bangladeshi population and B71, but its presence in the Zambian population and many wheat-infecting isolates outside the pandemic clonal lineage, may suggest the loss of *Art1_WB_ZM* in the known members of the South American population of the clonal lineage as well as the Bangladeshi isolates. Given also the presence of polymorphic variants of *Art1_WB_ZM* amongst other lineages of blast fungus, such as the *Oryza* and *Brachiaria* lineages, it may be the case that this is an effector present across multiple blast fungus lineages, but which happened to be lost in B71 and Bangladeshi wheat-infecting isolates, possibly to evade resistance in the hosts grown where B71 is present and in Bangladesh. It might also be that the observed presence/absence patterns of *Art1_WB_ZM* throughout blast fungus lineages is a result of a general absence of *Art1_WB_ZM* in the clonal lineage, but at some point there was a transfer of *Art1_WB_ZM* from other wheat-infecting isolates to the Zambian population. In the second case, it is anticipated that indicators of inter-lineage transfer of the region around *Art1_WB_ZM* would be evident.

In this work, I showed that the surrounding region is unique within the Zambian population, but I did not conduct extensive analysis across a range of blast fungus lineages to determine presence/absence patterns of this region. As more high quality assemblies become available, this work would become more likely to lead to identification of any sources for inter-lineage transfer of the effector candidate. The hypothesis that the presence/absence pattern of *Art1_WB_ZM* is due to selective loss in the non-Zambian populations in the B71 lineage may be more likely because of the reduced number of events to account for.

There is reason to believe that there has been a conserved replacement of the chromosome end in Zambian isolates which contains *Art1_WB_ZM*, corresponding to Contig19 in isolate ZMW20_04, separating the Zambian population from both B71 and the Bangladeshi isolates which have a different sequence. This is because the contig end region is absent in non-Zambian members of the B71 lineage, and appears to have a conserved region which replaces it in both Bangladeshi isolates and B71 at the start of B71 chromosome 2. Due to the presence of *Art1_WB_ZM* in wheat-infecting blast fungus isolates outside the pandemic clonal lineage, it could be that this regional presence/absence polymorphism occurred due to selection pressure from a deleterious impact Art1_WB_ZM has on pandemic clonal lineage isolates present in South America and Bangladesh. This could then have resulted in the loss of that region and its replacement by the current start of B71 chromosome 2. Additionally, this hypothetical selection pressure could instead be a result not from *Art1_WB_ZM*, but from another gene on the same Zambian-specific genomic region.

The attempt to identify a toxicity role using agroinfiltration found no evidence of toxicity

The possible results of the agroinfiltration experiment include lack of cell death response due to the effector candidate having no effect in *N. benthamiana* whilst also being undetected. Alternatively, the results could be induction of cell death in the plant due to it carrying an *R* gene able to detect Art1_WB_ZM, or cell death in the host resulting from the effector causing toxicity whilst not being detected. Of these, the plant carrying an effective *R* gene is unlikely, whilst Art1_WB_ZM having no impact is probable due to the experiment utilising *N. benthamiana*, itself distantly related to wheat and not a blast fungus host species. This experiment was limited in scope, but no toxicity impact was detected during this experiment, with the most likely explanation being that Art1_WB_ZM is not detected by any *R* gene in *N. benthamiana*, and also does not directly cause toxicity in *N. benthamiana*.

Infection assays did not indicate any impact from Art1_WB_ZM-deletion

The infection assays carried out did not indicate any direct impact on virulence occurring through deletion of *Art1_WB_ZM*, when carried out on a range of wheat cultivars from around the world. However, wheat cultivars known to be grown in South America, Bangladesh or Zambia were not available for use in these tests so it is possible that adaptation to host plants grown in one or more of these regions has occurred, but could not be detected with the experimental setup used here.

The inability of BTMP-S13-1 to reliably infect the Golden Promise barley cultivar was surprising given previous work within the lab (Barragan *et al.*, 2022) which used BTMP-S13-1 as a positive control, along with the same set of cultivars used here: Golden Promise, Nigrate, Fielder and Chinese Spring. In that previous work, BTMP-S13-1 was clearly shown to infect on Golden Promise. The positions on the Golden Promise leaves in this experiment did show marks on the leaves corresponding to the BTMP-S13-1 positions. As a consequence, it is possible that the specific BTMP-S13-1 isolate used here may have lost the ability to infect this cultivar at some point when it has been grown over time within the lab. As such, it may be interesting for future work to determine the ability of BTMP-S13-1 from different sources and grown under different conditions to infect on Golden Promise.

The AlphaFold3 prediction has increased confidence compared with the AlphaFold2 prediction

A useful local confidence metric when assessing AlphaFold predictions is the predicted local distance difference test (pLDDT) value. Varying between 0 and 100, higher model confidence is indicated by values closer to 100. Values greater than 70 are considered to generally indicate the backbone is correct but sidechains may not be placed accurately in the prediction. It is important to note that pLDDT does not provide information on the expected accuracy of the large-scale protein structure prediction, only on a per-residue local scale.

The initial Art1_WB_ZM AlphaFold prediction which was used in this work suffers from low pLDDT values over most of the structure. When it became available, AlphaFold3 was used to predict the Art1_WB_ZM structure, and the result displays significantly increased pLDDT values, corresponding to a greatly increased prediction confidence. Unfortunately by the time this tool became available, most of the project work had already been completed, and therefore future work on this effector candidate would be advised to use AlphaFold3 for structural prediction.

For assessing prediction confidence on a large scale over a protein structure, it is more useful to use the predicted aligned error (PAE) value, since this reflects the degree of confidence in predicted position between two residues. High PAE values indicate a predicted error which is higher between those two residues, and therefore a low confidence in their relative positioning. This pairwise measure is often displayed as a heatmap. The AlphaFold2 PAE values were generally high between residues at any distance from each other over the structure, whilst the AlphaFold3 prediction showed that most of the structure had low PAE values. This indicates that prediction confidence in the large-scale structure was low for the AlphaFold2 model, but significantly increased for the AlphaFold3 structure. The AlphaFold3 model also differs further from the HopU1 structure, and further work would be needed to understand the structural similarity of Art1_WB_ZM with members of the ART family.

Chapter 5: Investigation of an effector candidate present in Bangladeshi wheat blast isolates

Introduction

As discussed in Chapter 3, I identified genomic regions unique in Bangladeshi isolates which were absent in the South American B71 isolate. I then translated and predicted likely secreted proteins, which resulted in identification of an effector candidate, *APiasL3*. This has sequence similarity to a known effector in rice-infecting *M. oryzae* called *AVR-Pias*. *APiasL3*, within the pandemic clonal lineage, was present only in the Bangladeshi isolates, and absent in B71 and the Zambian isolates. This striking differential presence of the effector candidate within the B71 lineage may indicate a role in adaptation of the pathogen to the host population within Bangladesh. This is particularly interesting due to the Bangladeshi population forming the first sustained presence of wheat-infecting blast fungus outside South America since its emergence.

I also described in Chapter 3 how mini-chromosome contigs were identified in the three high quality Bangladeshi assemblies. As discussed in previous work, the “two-speed genome” may allow mini-chromosome carrying blast fungus isolates to separate the essential genes carried in the core genome from the more rapidly evolving mini-chromosome and thus reduce the risk of lethal genomic alterations whilst allowing faster adaptation (Dong, Raffaele and Kamoun, 2015). It has also been demonstrated that mini-chromosomes can facilitate the return of virulence on a resistant host through loss of a mini-chromosome containing avirulence genes, as in the case of rice-infecting blast fungus isolates carrying *AVR-Pik* variants infecting on *Pik*-containing hosts (Kusaba *et al.*, 2014). In blast fungus isolates, it has also previously been identified that structural rearrangements can take place between the mini-chromosome and core genome, and that these can involve avirulence genes (Langner *et al.*, 2021).

In this chapter, I discuss my use of AlphaFold2 to predict a structure for APiasL3, followed by my attempts to generated *APiasL3*-deletion transformants for subsequent use in infection assays on wheat. The transformants proved unsuccessful, possibly due to the presence of *APiasL3* on the mini-chromosome. Through adapting the pipeline discussed in Chapter 3, I was able to identify that *APiasL3* belongs to a larger gene family distributed across blast fungus lineages, which includes the previously identified *AVR-Pias*. I was then able to use the predicted APiasL3 structure for downstream analyses such as identification of conserved residues across the gene family. The 35A residue, in particular, was predicted to be under positive selection by multiple selection analysis techniques.

Aims

The aim of this chapter is to investigate the effector candidate, *APiasL3*, present within the Bangladeshi isolates, and determine if this is a functional effector aiding Bangladeshi wheat blast fungus to better infect its host. I used AlphaFold2 to predict the APiasL3 structure and allow downstream investigations making use of the effector candidate structure, before analysing existing field-collected transcriptome data from the 2016 infection season in Bangladesh to determine if there is reason to believe *APiasL3* is expressed under natural

conditions. Next, I attempted to generate *APiasL3*-deletion transformants in a similar manner to that described in chapter 4, with the goal of performing infection assays to assess any differential ability to infect compared to the wild type. I also adapted the pipeline used in the work discussed in chapter 3 for mapping presence and absence of effector candidates across blast fungus lineages to identify sequence-similar proteins present across blast lineages, to attempt identification of a protein family APiasL3 is a part of. Finally, I then used this information to conduct selection analysis, identify conserved residues and used further AlphaFold predictions to investigate a hypothesis around potential APiasL3 targets.

Results

The AlphaFold prediction of the APiasL3 structure is not high-confidence

As was done with Art1_WB_ZM, I used the ColabFold-AF2 tool to generate a structural prediction for APiasL3, as can be seen in Figure 24, panel A. This prediction of the structure shows three alpha helices (marked a-c) connected by short unstructured regions, and the three helices are arranged in a loosely parallel structure. The confidence in this prediction was low, as indicated by the generally low pLDDT values shown superimposed on the structures in panel B. Panel C displays the pLDDT values per residue for this structure, with a significant number of those residues having values under 50, signifying poor confidence in the prediction.

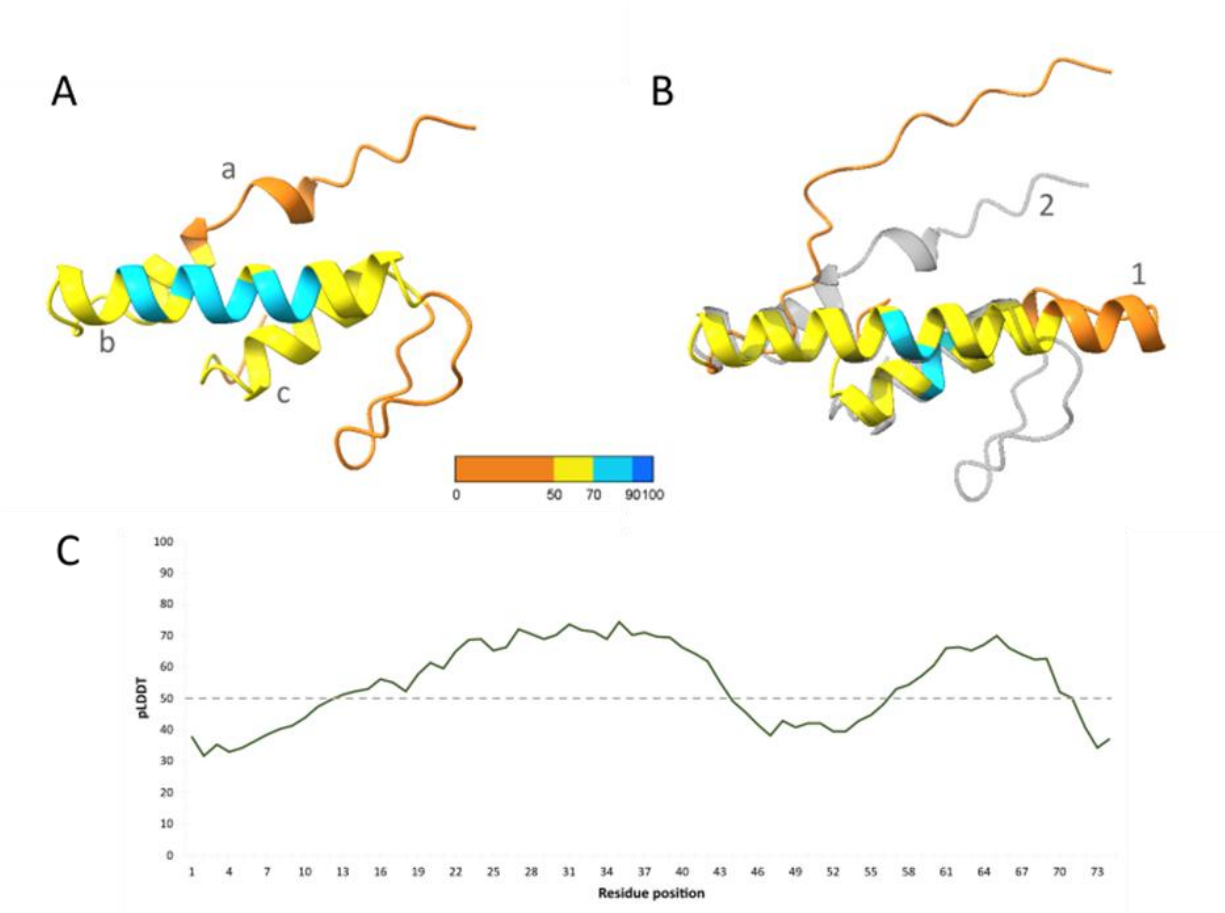

Figure 24. (A) Structural prediction on APiasL3 using ColabFold-AF2, composed of three major alpha helices (marked a, b and c). (B) Overlapping structural prediction of APiasL3 using ColabFold-AF2 (the shorter main alpha-helix is marked with '2', and the entire structure is grey and semi-transparent) and AlphaFold3 (longer main alpha-helix, marked with '1'), both coloured by pLDDT values where blue indicates high confidence and orange and yellow

are lower confidence. (C) pLDDT values for the AlphaFold2 structure prediction shown in panel A and structure 2 in panel B. A considerable number of residues have a pLDDT value below 50, signifying poor confidence in the prediction.

Figure 24, panel B shows the AlphaFold3 structure coloured by pLDDT value superimposed over the AlphaFold2 structure in grey. It illustrates that when using AlphaFold3, the main alpha helix is longer (the helix next to '1', and corresponding to helix 'b' in panel A), but with a region of lower confidence shown in orange. Additionally, the alpha helix marked 'a' in panel A is not present in the AlphaFold3 prediction, and is instead predicted as a long, low-confidence strand. The long strand connecting helices b and c in panel A is also significantly shorter in the AlphaFold3 structure, being absorbed into the alpha helices. It is not clear that the AlphaFold3 model has any greater confidence than the AlphaFold2 model, unlike the predictions for Art1_WB_ZM. Throughout this work, the AlphaFold2 prediction was used due to AlphaFold3 only becoming available towards the end of the project.

APiasL3 is located on both the core chromosome and mini-chromosome in different Bangladeshi isolates

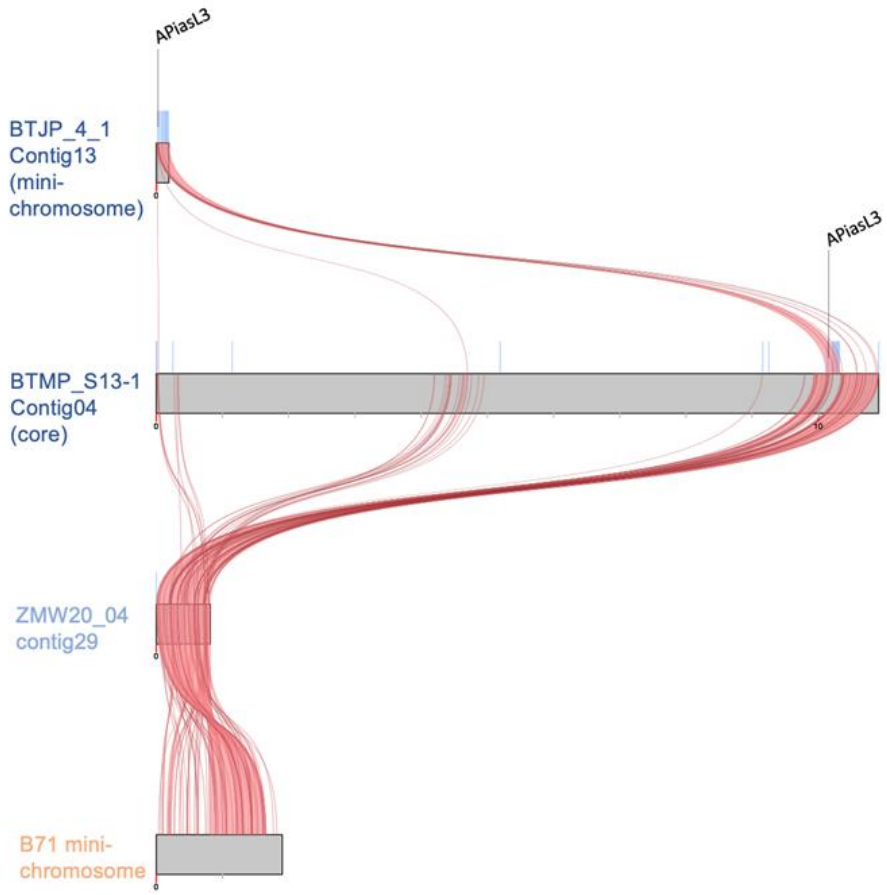

Figure 25. Genomic rearrangements and APiasL3 gene locations within B71 lineage genome assemblies. Red ribbons mark alignments between contigs, whilst blue bars over contigs mark regions absent in B71.

According to the analysis described in chapter 3, the mini-chromosome contigs of the Bangladeshi isolates BTMP_S13_1, BTJP4-1 and BTGP1-b were identified through mapping CHEF gel data to the genome assembly. Using BLAST, I identified the locations in each of these assemblies where *APiasL3* was found. As can be seen in Figure 25, Contig13 in BTJP4-

1 contains a copy of *APiasL3*, and this whole contig was also identified as sequence present in the mini-chromosome. The sequence of this whole contig was identified as being absent in the B71 isolate. This region, containing a copy of *APiasL3*, is also present in the BTMP-S13-1 isolate, however the assembly data suggests this entire region is present in the end segment of contig04, which is a core chromosome. The entire end segment of this contig (excluding the *APiasL3*-containing segment) aligns well with the B71 mini-chromosome (as well as sequence present in the Zambian isolates), however *APiasL3* is not present in complete form in either B71 or any Zambian isolate. I mapped nanopore sequences for isolate BTMP-S13-1 to its assembly for contig04, which did not provide any indications of an assembly error in this region, suggesting that this sequence corresponding to the mini-chromosome has indeed been integrated into the core genome in this isolate. It remains possible that there is a duplication of this region in the BTMP-S13-1 genome, complicating the analysis.

In order to test if *APiasL3* is expressed under natural conditions, I mapped transcriptome data from samples from the 2016 infection season in Bangladesh to the BTJP4-1 assembly. I found a transcript at the *APiasL3* locus in one of the four symptomatic plant samples, and not in the asymptomatic plant samples, supporting the hypothesis that *APiasL3* is expressed naturally.

Attempted generation of *APiasL3*-deletion transformants was not successful

I used the same method as was used in chapter 4 to generate 35 *APiasL3*-deletion transformants in the Bangladeshi isolate BTJP4_1. Of these, only five were able to grow on hygromycin-CM plates, and were designated as samples Bt8, Bt17, Bt30, Bt32 and Bt33. However, the final two (Bt32 and Bt33) had unusual phenotypes and were discarded from further analysis, leaving the final three high-confidence transformants Bt8, Bt17 and Bt30 for further analysis.

In an attempt to verify if the transformation process had succeeded, I conducted PCR checks using the same methods as were previously used for the *Art1_WB_ZM*-deletion mutants, but with the diagnostic primers matching the *APiasL3* locus. Despite applying multiple approaches, reagents and attempts to check for contamination, I was unable to obtain conclusive results due to multiple problems, which included diagnostic region primers appearing in control samples lacking the diagnostic region. I eventually shifted to using sequencing to obtain a better understanding of the transformant genomes in this region.

For whole genome sequencing I followed the same procedure as used for the sequencing preparation of the Zambian transformant, and obtained sufficient concentration and purity values using the same QIAGEN DNA extraction kit to allow for Illumina sequencing (see Table 5).

Table 5. DNA concentration and purity values used for whole genome sequencing of BTJP4-1 transformant samples.

Sample	**Concentration (ng/microlitre)**	**260/280**	**260/230**
Bt8	22.59	1.81	1.25
Bt17	19.87	1.90	1.63
Bt30	27.36	1.85	1.22

As shown in Figure 26 for sample Bt8, this approach identified coverage for both *APiasL3* and the hygromycin marker gene, although a greater read count was obtained by mapping against the simulated transformant sequence. As shown in the figure, mapping to the whole genome and only to contig13 (the transformation target) suggested the presence of sequence for both *APiasL3* and the hygromycin resistance gene. Samples Bt17 and Bt30 showed similar mapping results, and this method therefore did not imply that *APiasL3* had been completely removed from any of these transformant attempts.

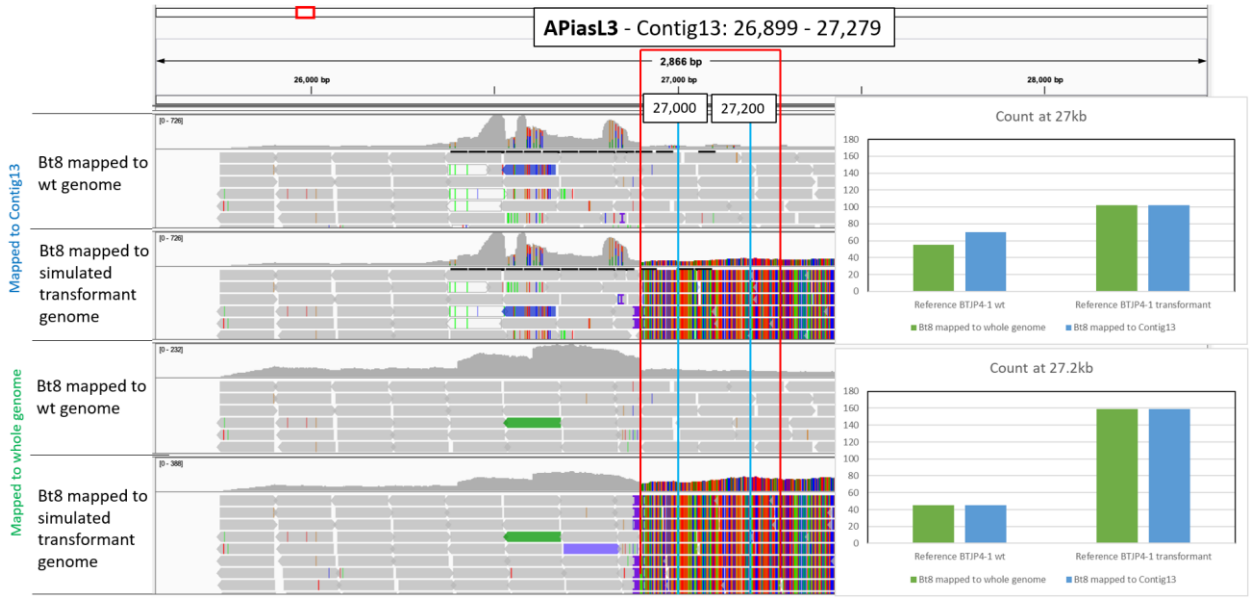

Figure 26. Bangladeshi transformant Bt8 reads mapped onto BTJP4_1 wild type and simulated transformant genomes. The read count is shown in inset graphs at 27kb and 27.2kb to demonstrate that reads map to both the wild type and simulated transformants over the whole region. Image created by author using IGV (version 2.17.0), and annotated for clarity.

I performed a leaf-drop infection assay to attempt to verify any impact on virulence from *APiasL3*-deletion in BTJP4_1 (Figure 27). A summary of the results is also shown in Table 6. Bt8, Bt17, and Bt30 (the three *APiasL3*-deletion transformants selected for analysis) appear to cause similar infection symptoms to the wild-type isolate, BTJP4_1. Similarly to in the Zambian isolate infection assay, BTMP_S13_1 (control isolate) did not infect on several leaves of the Barley cultivar, Golden Promise.

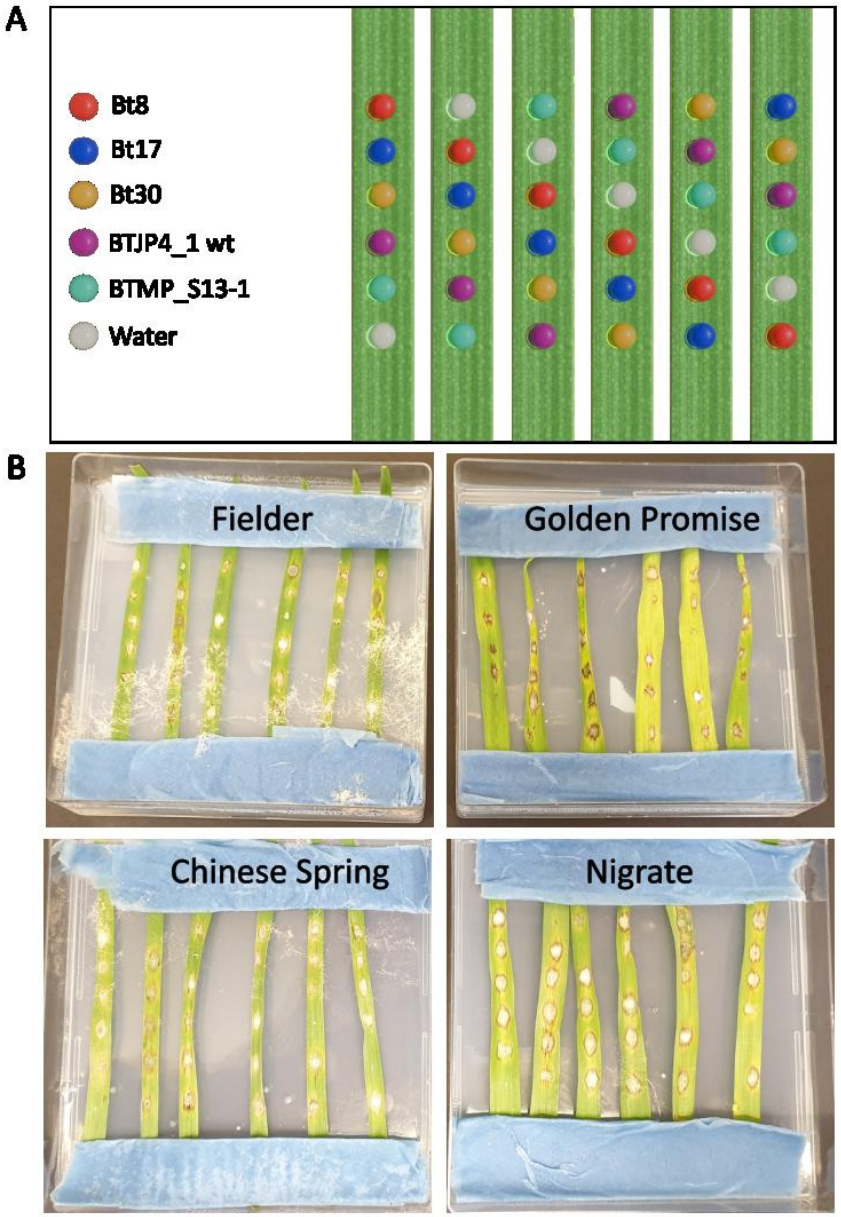

Figure 27. Panel A illustrates the setup of the infection assay using two wheat cultivars (Chinese Spring and Fielder) and two barley cultivars (Nigrate and Golden Promise). This used BTJP4_1 (wild type), three APiasL3 deletion transformants of the same isolate, a water control and the isolate BTMP-S13-1 as a further control isolate. Images were taken 4dpi. On each plate, each leaf from left to right has had the drops displaced one position downwards, with the bottom-most drop being moved to the top, which is illustrated in the top legend (produced using the Blender 4.0 3D modelling software). Panel B shows the resulting lesions after performing the experiment

Table 6. Results from initial infection assay. Each '+' indicates one resistance spot on a leaf, and each '-' indicates an absence of a visual response.

		Cultivar			
Isolate		Fielder	Chinese Spring	Golden Promise	Nigrate
	Bt8	++++++	++++++	++++++	++++++
	Bt17	++++++	++++++	++++++	++++++
	Bt30	++++++	++++++	++++++	++++++
	BTJP4_1 (wild type)	++++++	++++++	++++++	++++++
	BTMP_S13_1 (Control)	++++++	++++++	--++-+	++++++
	Water (Control)	- - - - - -	- - - - - -	- - - - - -	- - - - - -

APiasL3 is a member of a family of proteins related to *AVR-Pias*, distributed across blast fungus lineages

In chapter 3, I described the process of identifying *APiasL3* in multiple B71 lineage blast isolates from Bangladesh, through the use of a custom pipeline utilising BLAST searches. In order to allow for detection of similar protein sequences, I expanded this pipeline to perform recursive BLAST searches, taking the outputs from the first search as inputs for a second round of BLAST searches using the same genome dataset for the database.

Through conducting a recursive search using this modified pipeline, I identified that *APiasL3* appears to be part of a larger family of proteins, with presence across a range of host-specific blast fungus lineages, including the *Oryza*, *Triticum*, and *Lolium* lineages. An output heatmap is shown in Figure 28. This shows that many other members of this family are present across different blast fungus lineages.

These different sequences were classified as separate genes within this family when the percentage identity between a hit and the query fell below 90%. *APiasL2* is present in a small number of *Oryza* and *Eleusine* isolates. *APiasL4* is present in *Stenotaphrum* and *Brachiara* isolates. *APiasL5* is carried by isolates in the *Oryza*, *Setaria, Brachiaria, Stenotaphrum, Triticum, Lolium* and *Eleusine* groups, and is also the only family member present in all B71 lineage isolates, and indeed the only one present in all wheat-infecting isolates. *APiasL6* was only identified in the two *Brachiaria* isolates, whilst *APiasL7* was identified only in one *Eleusine* isolate. *APiasL8* is present only in the *Digitaria* isolates, and *APiasL9* was found in the single *Pennisetum* isolate only.

APiasL3 occurs in the Bangladeshi population, as shown in Figure 28, but not in other wheat-infecting isolates. Additionally, variants of the *APiasL3* gene with some polymorphisms are present in other blast fungus lineages, such as those infecting *Brachiaria*, *Setaria, Eleusine*, and one isolate infecting *Lolium*. A variant with 97-98% identity is carried by many rice-infecting lineage isolates, and one of these *Oryza* isolates was also identified carrying the same sequence as is present in the Bangladeshi isolates. Some of the *Oryza* isolates carry a second copy of *APiasL3*, but no other lineage of blast fungus appears to have isolates with a second copy.

During the analysis of the results of this recursive BLAST search, it became apparent that *APiasL3* has 68.9% identity with a previously-identified avirulence gene found in the *Oryza* blast fungus lineage, called AVR-Pias. Previous work (Shimizu *et al.*, 2022) identified that the NLR pair Pias-1 (helper) and Pias-2 (sensor), can detect AVR-Pias, and that resistance depends on both Pias-1 and Pias-2. In Figure 28, I show that this pipeline identified *AVR-Pias* as being present in only two of the rice-infecting isolates used in this work, but there are also variants of *AVR-Pias* present in *Setaria, Brachiaria, Lolium, Eleusine* and one *Triticum* isolate from outside the B71 lineage. Whilst having multiple copies of *AVR-Pias* appeared to be rare, one *Brachiaria* and one *Eleusine* isolate had a second copy with a polymorphic sequence to the first copy.

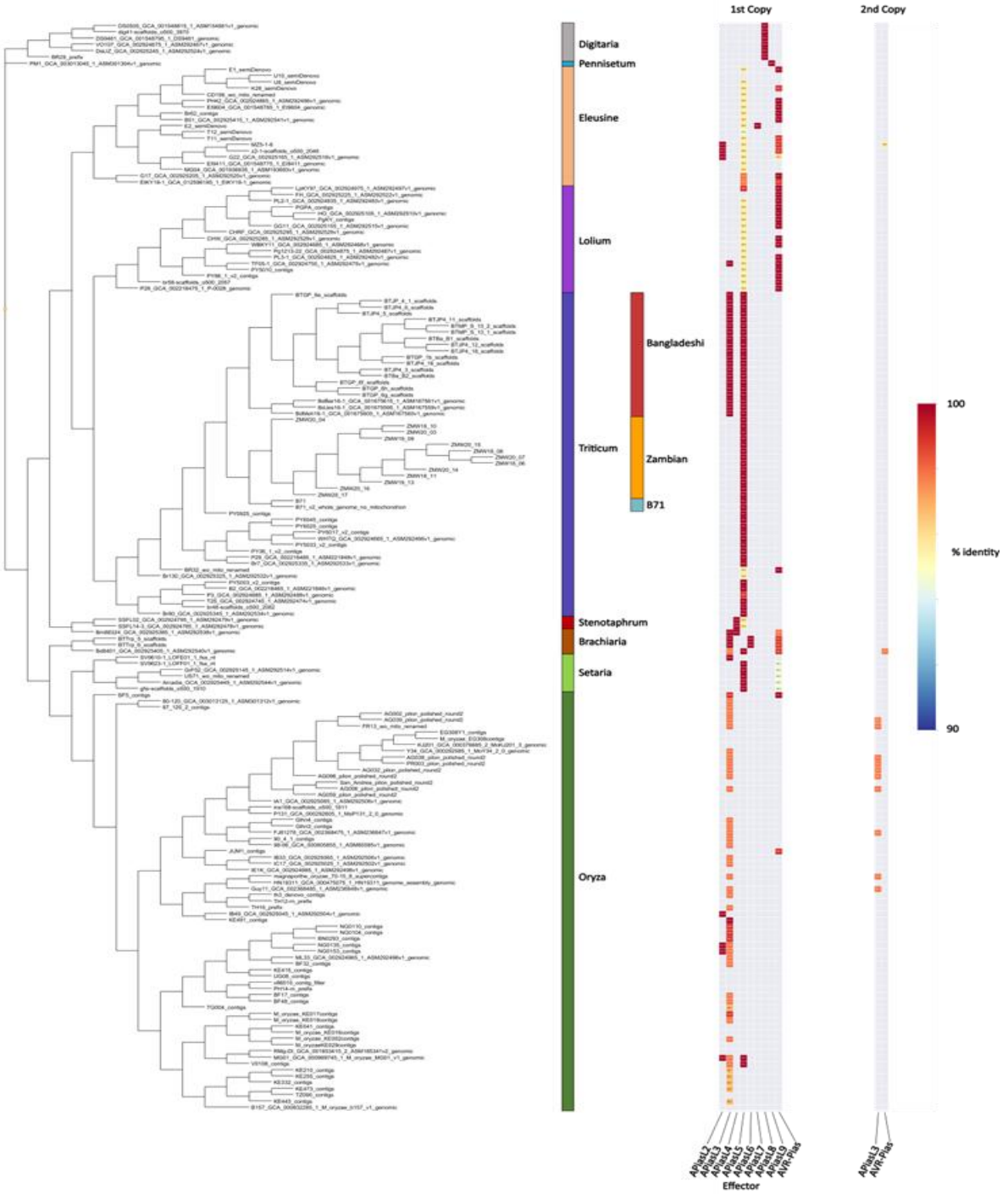

Figure 28. APiasL family effector candidate presence/absence distribution throughout blast fungus lineages, where the cladogram was generated using the kSNP3 software. Each effector candidate is present on a different column, and the heatmap displays the % identity of each hit, determined via BLAST, if above 90% identity. The location of Bangladeshi and Zambian isolates is highlighted. Column order for effector candidates (1st gene copy) is: APiasL2, APiasL3, APiasL4, APiasL5, APiasL6, APiasL8, APiasL9, AVR-Pias. The 2nd gene copy column order is: APiasL3, AVR-Pias.

The degree of sequence similarity for the non-redundant members of the *APiasL* family can be seen in Figure 29, with the *APiasL3* branch at the top and *AVR-Pias* in the middle.

The sequences with the longest branches are *APiasL8* and *APiasL9* near the bottom of the tree, which are also the only ones present in the *Digitaria* and *Pennisetum* lineages respectively.

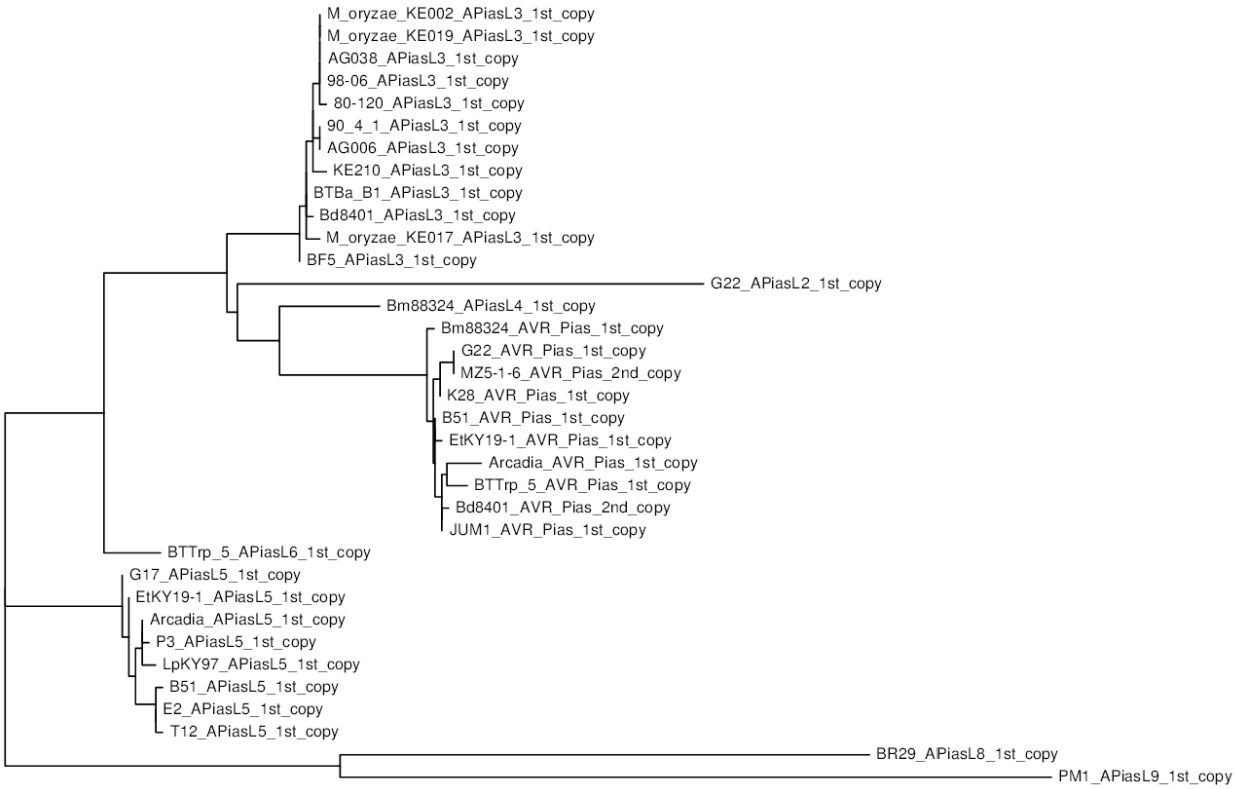

Figure 29. Tree of non-redundant APiasL family members. The branch containing APiasL3 is at the top of the figure, with the AVR-Pias branch in the centre. Tree structure generated using MEGA v11.0.13.

$\frac{d_N}{d_S}$ analysis of the *AVR-Pias* family identified potential positive selection in APiasL3

The $\frac{d_N}{d_S}$ ratio is a measure of selection pressure for protein-coding sequences which uses non-synonymous (which alters the amino acid sequence being coded for by a gene) to synonymous mutations (a mutation in the gene which does not alter the amino acid sequence being coded for) (Kryazhimskiy and Plotkin, 2008). Due to the potential impact from selection pressures which occur with non-synonymous mutations, this ratio is used as a measure for selection acting on a protein, and can be applied to the residues within a protein. A ratio greater than one is typically understood to mean positive selection is acting on a residue, a ratio less than one indicates purifying selection, and a value of one indicates no driving selection. Purifying selection means that protein sequence changes are inhibited, whilst positive selection acts to promote sequence change. As such, on the level of individual residues, one could identify those amino acids which are conserved throughout a protein family and yield deleterious results when substituted, and one could identify those residues which are prone to substitution.

In an effort to check for positive selection at any interface between APiasL3 and an interacting protein, I tested two hypotheses with the phylogenetic analysis software PAML. I first used the branch model to test a null hypothesis (assuming the same $\frac{d_N}{d_S}$ ratio for all branches in the protein family, and across all residues in each protein sequence) against the alternative hypothesis with one ratio across just the *APiasL3* branch, and a second background $\frac{d_N}{d_S}$ ratio for the other branches in the *APiasL* family. For the APiasL3 branch of the *APiasL* family (as shown in Figure 29), I obtained a $\frac{d_N}{d_S}$ ratio of 0.87. This was tested using both the one ratio ($\omega_0 = 0.65$) and two ratio ($\omega_0 = 0.63$) models. The null hypothesis could not be rejected at a 5% significance level, so it cannot be concluded that there is evidence of the *APiasL3* branch having a higher $\frac{d_N}{d_S}$ ratio than the background level for the *APiasL* family. It is known that the branch model will only show a high value for the $\frac{d_N}{d_S}$ ratio if the majority of residues in the protein are under positive selection, which is rare. I conducted a more sophisticated, second test using a branch-site model test, identifying two sites preliminarily as being under positive selection: 35 A (62.3%) and 41 V (92.3%).

I used the SLAC (Single Likelihood Ancestor Counting) method from the HyPhy package (Pond, Frost and Muse, 2005; Kosakovsky Pond *et al.*, 2020) on the *APiasL* family. It highlighted the 41V and 35A amino acids which were identified by PAML, as also being positively selected, but with additional codons also identified as having higher $\frac{d_N}{d_S}$ ratios. Additionally, the HyPhy FEL method (Fixed Effects Likelihood) highlighted three positively (17C, 35A, 57R) and multiple negatively selected amino acids. In the set of analysed isolates, there were no synonymous mutations for both residues 17C and 57R, however, rendering the conclusion of them being under positive selection invalid. This combination of methods seem to support 35A as being positively selected through multiple analyses, but further investigation of this is required.

I conducted a search using the AlphaFold2 APiasL3 structure prediction as a query against the VAST structural database (Gibrat, Madej and Bryant, 1996), in an attempt to identify similar protein structures. The greatest similarity was to an entry for crystal data of RanGAP2-WPP bound to the Rx-CC domain, where the predicted APiasL3 structure matched most closely to the RanGAP2-WPP domain, shown in Figure 30. Rx is a is a sensor CC-NLR found in *Solanum tuberosum* that detects the coat protein of Potato virus X, facilitating resistance (Bendahmane *et al.*, 1995). Only three residues were shared between APiasL3 and the WPP domain: 31E, 35A, 39I, all of which are predicted to sit on the same side of the same alpha helix in adjacent positions, which in the RanGAP2-WPP domain sits next to or at the interaction site. One of these, 35A, was identified by PAML as likely to be under positive selection, in addition to being highlighted by the SLAC and FEL methods.

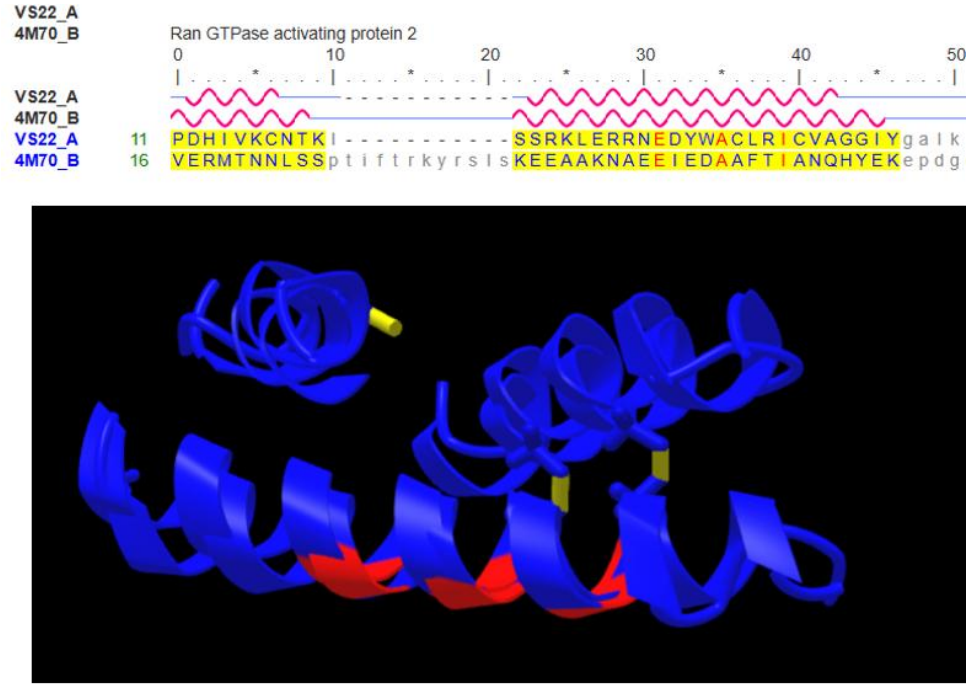

Figure 30. Alignment (above) and structure (below) overlay of APiasL3 AlphaFold2 prediction and RanGAP2-WPP crystal data. Residues conserved between the two sequences are marked in red. Figure created with the use of the VAST web tool.

Previous experimental work has indicated that Rx coexpressed with RanGAP2 displayed improved resistance to Potato Virus X. RanGAP2-WPP domain coexpression with Rx instead demonstrated reduced resistance (Tameling *et al.*, 2010). As such, I wished to explore if APiasL3 could potentially be mimicking the RanGAP2-WPP interaction with Rx, and binding to the CC domain of an NLR present in current hosts or in related grasses, such as *Lolium*, and if this could play a role in the ability to infect different host species.

In an effort to investigate further if this structural similarity could inform on potential APiasL3 interactions, I used AlphaFold2 predictions, providing two sequences of potential interaction partners. Acting as a control and to verify that prediction on this type of interaction is possible, I used Rx-CC and RanGAP2-WPP sequences in the AlphaFold2 prediction, and it successfully replicated the crystal data structures with a significant closeness to the experimental data. When I attempted to predict any interaction between APiasL3 and Rx-CC, however, the pLDDT value was less than 50 for most of the length of APiasL3 for all predicted structures, and therefore no interaction could be viewed as likely based on this prediction.

Since Rx is not present in blast fungus host plants, I used BLAST searches to attempt to identify any similar sequences which might be present in blast fungus hosts. The closest match was the RGA5-like XM_051329199.1, which had only 29% identity to Rx, and AlphaFold2 predictions of interaction of this whole protein with APiasL3 resulted in pLDDT values between 20-40 over the length of APiasL3, which does not indicate interaction is likely. Further, AlphaFold2 predictions of APiasL3 with only the CC domain of XM_051329199.1 gave the most likely prediction to occur in a geometry which is not physically possible.

Because amino acids conserved across a protein family may be indicative of a critical structural or functional role, I used the tool Consurf to screen for such conserved residues across the APiasL family, and map them onto the predicted structure of APiasL3, as shown in Figure 31.

Two conserved cysteine residue pairs are predicted to be situated in close proximity, which may enable them to form two disulfide bonds binding together two of the alpha helices in the predicted AlphaFold2 APiasL3 structure. Both of these disulfide pairs are conserved across the entirety of the family. In addition, the conserved cysteine residues across the family imply that a 3rd disulfide bond could form only in the APiasL3 branch, along with APiasL4, which would potentially bind to the third predicted alpha helix. Thirty seven percent of the APiasL family protein variants have this third cysteine pair (17C, 72C), but only one of these cysteine residues (17C) is also present in the APiasL5 branch, and is therefore present in around 63% of the family. This 17C residue was highlighted in the FEL analysis to possibly be under positive selection. It is notable that AVR-Pias lacks either of the cysteine residues in this pair, despite being more closely related to APiasL3 than APiasL5.

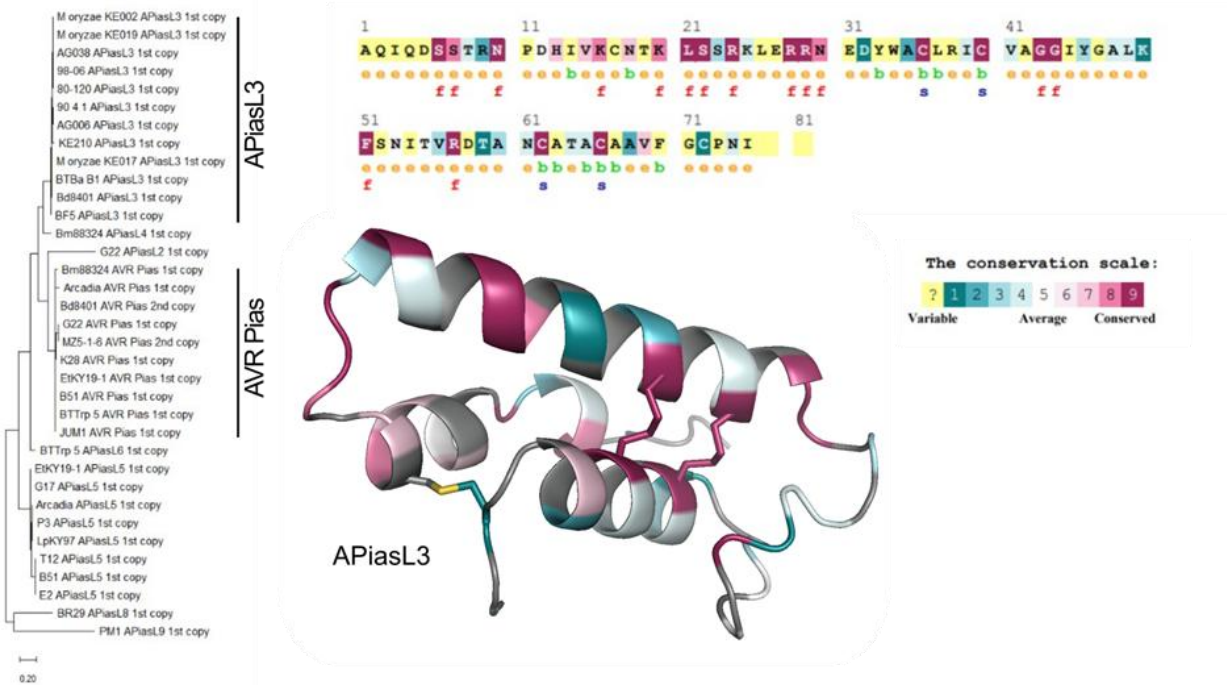

Figure 31. Results from Consurf analysis of the APiasL family, showing conserved and variable residues from the family mapped onto the APiasL3 structure. The APiasL3 sequence is also shown, with the variability or conservation of each residue throughout the family represented in the same colour coding. Purple residues are highly conserved across the family, whilst those shown in blue are variable. There are two pairs of cysteine residues which are fully conserved throughout the family, whilst a 3rd pair which could form a disulfide bond is only present in the APiasL3 and APiasL4 branch.

Of the three residues which are conserved between APiasL3 and the RanGAP2-WPP domain, 35A (also highlighted by the $\frac{d_N}{d_S}$ analysis) is notable because the Consurf analysis reveals it to be highly variable within the APiasL family. The 39I residue is also slightly variable, as was the 41V residue, highlighted by the $\frac{d_N}{d_S}$ analysis.

Discussion

APiasL family distribution throughout blast fungus lineages

Different members of the *APiasL* family are present across the full breadth of blast fungus lineages surveyed, as shown in Figure 28. *APiasL5* is present across approximately 50%

analysed blast fungus isolates, and in particular in 100% of the wheat-infecting isolates, and nearly all *Lolium* and *Eleusine* lineage isolates. In contrast, *APiasL7* and *APiasL9* were present in only one isolate.

One characteristic this analysis reveals of many *APiasL* family sequences is that they are often present across only some isolates within a lineage, and may also be present in one or more other lineages which may not be closely related. For example, *APiasL2* is present in multiple isolates from the *Eleusine* lineage and the *Oryza* lineage, both of which are relatively distantly related, and there is no evidence it is present in any of the other more closely-related lineages. Another example of this is *APiasL3*, which is present in the Bangladeshi subset of wheat-infecting isolates, but is also present in other lineages such as one *Lolium* isolate which are closely related to the *Triticum* lineage. Meanwhile, *APiasL3* is present across the *Brachiaria* isolates included in this work and an *APiasL3* variant is present in many rice-infecting isolates which are more distantly related to the *Triticum* lineage.

It is possible this pattern of partial presence across multiple more distantly-related isolates could be due to transfer of material between two distantly related lineages. This could be linked with potential mini-chromosome transfer between different blast fungus lineages. Alternatively, the different members of the *APiasL* family which are present across these less-related lineages, such as *APiasL2*, may have once been more widespread across lineages but were subsequently lost in other lineages due to it conferring a disadvantage during infection on that host plant. Given the presence of one or more *APiasL* family sequences in every blast fungus lineage analysed, it is likely that the *APiasL* family has diversified from an initially relatively homogenously present ancestor which later specialised to aid in infection, or avoid detection, on different host plants targeted by each blast fungus lineage. If these *APiasL* family members confer advantages during infection, then diversification is likely linked with the emergence of certain host-specific lineages, or infection on a new host. Following this, the distinctive patterns of presence of family members in distantly-related lineages but absence in more closely-related populations, such as the *Brachiaria* isolates and Bangladeshi wheat-infecting isolates in the case of *APiasL3*, indicates either subsequent loss in non-Bangladeshi wheat-infecting isolates, or transfer between the two distantly-related populations. In future work, having access to a greater number of isolates from some of the poorly-represented lineages would be beneficial to determine the true prevalence of these sequences amongst the different lineages and may provide insight into their origins.

Transformant generation attempt and infection assay

The *ApiasL3*-deletion transformant attempt may have failed due to the locus of *APiasL3* lying on the mini-chromosome. This may have the effect of allowing genomic rearrangements to occur during the transformation process, introducing duplications of the *APiasL3* locus or surrounding region. Of these, only some might be correctly transformed, thus leading to the detection of both *APiasL3* and the Hygromycin-resistance gene within the transformant attempts. Another possibility is that there are undetected duplications of the mini-chromosome sequence containing *APiasL3* within the genome.

I performed the infection assay in parallel with assessing the sequencing results for the three Bangladeshi transformant attempts, and consequently it was not determined that the transformation was unlikely to have succeeded until the infection assay had been performed.

It is likely that at least one copy of the effector candidate remains in these transformants, and therefore any infection assay undertaken with these samples cannot be relied upon to show impacts from *APiasL3*-deletion. Regardless of this, no differential infection characteristics were noted between the Bangladeshi transformants and the wild type isolate.

As previously discussed, the ability of the mini-chromosome to facilitate genetic material transfer into and out of the core genome is a mechanism able to aid the process of adaptation in pathogen genomes. It is notable that *APiasL3* lies within the genomic region that in BTMP-S13-1 was incorporated onto the end of a core genome contig (contig 4), whilst it appears to remain in the mini-chromosome on the other two Bangladeshi isolates with high quality assemblies used in this work. Whilst BTMP-S13-1 was not used for transformant generation here due to this unusual character of its genome, it may be that *APiasL3*-deletion transformation attempts would be more likely to succeed using this isolate. Alternatively, the significant mini-chromosome read coverage seen on the end of BTMP-S13-1 contig 4 in Figure 8 may indicate that this genomic region contains duplicated sequence still present in the mini-chromosome, and that as such, *APiasL3* is present in both the mini-chromosome and the core genome, and that transformation may still fail.

If *APiasL3* is present both in the core genome and mini-chromosome within one of the Bangladeshi isolates, this may be an ongoing demonstration in a new disease outbreak of how novel effector functions are developed in a rapidly-evolving genomic compartment whilst retaining existing function in the core genome, facilitating adaptation to the new environment.

Cysteine pairs are conserved in the *APiasL* family

If it is the case that two pairs of cysteine residues are fully conserved across the *APiasL* family, and if the true structures of the members of the *APiasL* family do allow for formation of these disulfide bonds, this may indicate an important role for the protein structure in its functioning. These could be acting to stabilise the protein. In the predicted APiasL3 structure, it appears that two of the alpha-helices constituting the structure would be held more strongly in place by the two fully conserved cysteine pairs, whilst the third alpha-helix segment would be similarly held more strongly to one of those original helices by the 3rd cysteine pair which is not conserved throughout the APiasL family and could only occur in APiasL3 and APiasL4.

Given the presence of only one of these cysteine residues in APiasL5, which is more distantly related to APiasL3 than AVR-Pias, which lacks either cysteine residue, this may indicate that the cysteine pair was originally prevalent, but the ability to form the disulfide bond has been lost in the other members of the APiasL family, perhaps to evade detection or enhance virulence due to altered properties or behaviour of the protein. Since the Consurf analysis incorporates structural information into its prediction, experimental structural data, or at the least, higher confidence predictions such as those from newer structure prediction tools rather than AlphaFold2 would be valuable to improve the results of similar future analysis.

Selection analysis in the *APiasL* family

In order to test for residues under selection pressure, with the ultimate goal of identifying an interaction interface, I used several methods to attempt to determine the $\frac{d_N}{d_S}$ ratio for APiasL3 within the APiasL family. When using the branch model with the PAML software I was not able

to conclude that APiasL3 had a higher $\frac{d_N}{d_S}$ ratio than the rest of the APiasL family. This is expected since only in unusual cases where most residues are under positive selection would the branch model return a result indicating positive selection. I also used the branch-site model, which identified the 35A and 41V residues as potentially being under positive selection. The branch-site model has, however, been shown to indicate positive selection falsely on between 20-70% of simulated sequences (Zhang, 2004), with it being recommended to supplement the branch-site model with additional tests. More recent work has also suggested that the branch-site model has been misinterpreted in the literature multiple times (Kowalczyk, Chikina and Clark, 2021). In addition to the above tests, I used the SLAC method from the HyPhy package, which also identified those two residues amongst several others as being positively selected, whilst the FEL method from the HyPhy package also identified 35A as being under positive selection, amongst several other residues. These results should be considered as preliminary indications, and further work would benefit from the use of more advanced analysis methods and a greater sample of sequences as more genome assemblies become available.

Structural similarity analysis depends on a low-confidence structural prediction

My attempt to identify structurally similar matches to APiasL3 found similarity to the RanGAP2-WPP domain. Three residues (31E, 35A, 39I) were the same between these two sequences, all of which are predicted to sit adjacent to each other on an alpha-helix on the predicted APiasL3 structure. One of these, 35A, also happens to be predicted to be under positive selection by multiple selection analysis tools, as mentioned in the previous section. These may tentatively hint at the interaction site, but are only valid if the similarity to RanGAP2-WPP is robust.

Ultimately, these attempts depend on the accuracy of the structural prediction used in this work. As has been discussed previously, some doubt about the accuracy of the AlphaFold2 model is appropriate, and therefore, further work depending wholly on the accuracy of the APiasL3 structural prediction must be interpreted with caution. Ultimately, very low confidence could be assigned to the AlphaFold2 predictions of interactions between APiasL3 and the Rx-CC domain, and between APiasL3 and the most similar sequences to Rx I identified in grasses. Because of these issues, it is not possible to identify any likely target or interaction partner for APiasL3 based on the current information. Further work could systematically utilise sequences from a greater range of grass species, and should also consider the use of AlphaFold3 when predicting possible interactions due to its generally improved prediction confidence, or the use of other new structure prediction methods.

Chapter 6: General Discussion and Future Work

General Discussion

In this work I identified two effector candidates, *APiasL3* present in Bangladeshi wheat-infecting blast fungus isolates, and *Art1_WB_ZM* present in Zambian isolates. Whilst both populations of isolates belong to the B71 pandemic clonal lineage which has recently spread to three continents, within the B71 lineage these two effector candidates are exclusively present in only their population group and are absent in the rest of the B71 lineage. Such clear differential presence throughout a clonal lineage, such as might occur through selective gain or loss within the Bangladeshi or Zambian population compared to the other members of the B71 lineage, may indicate some advantage conferred by the candidates. Investigating these effector candidates through the use of computational techniques, and attempting to identify any virulence impact experimentally, was the primary justification for conducting this work.

The effector candidate identification was performed using a custom pipeline to first identify population-specific genomic regions compared to the B71 reference genome, and then using 6-frame translation and signal peptide prediction, followed by a BLAST-based approach to identify the effector candidates themselves. Whilst multiple hits were found, two were shortlisted for further study due to their sequence-similarity or predicted structural similarity to previously studied proteins. As sequence similarity databases and structural similarity methods and databases improve, it is possible that some of those other sequences which were not further studied in this work might be interesting for future work, although with enough resources, work on any of the population-specific candidate effectors could be promising.

The *Art1_WB_ZM* effector candidate is present in the core genome in Zambian isolates. Structural similarity was identified between the AlphaFold2 structure prediction and HopU1, an RNA-binding protein belonging to the ART family of proteins which is required for virulence in *Pseudomonas syringae*. Such structural comparisons, however, depend entirely on the accuracy of the structural prediction of Art1_WB_ZM, and I subsequently predicted the Art1_WB_ZM structure using AlphaFold3, obtaining a much higher-confidence structural prediction than was used in this work comparing similarity to HopU1 ART. I was able to produce successful effector-deletion transformants which I verified via sequencing. However, an infection assay using a panel of ten wheat cultivars did not identify any differential infection ability compared to the wild type. Cultivars widely grown within Zambia were not used in this experiment, and therefore further work is required before ruling out virulence impact.

The *APiasL3* effector candidate is present in mini-chromosome contigs in Bangladeshi isolates, but is present in a core genome segment of BTMP-S13-1 which appears to have been integrated from the mini-chromosome. My attempts to generate effector-deletion transformants for *APiasL3* were not successful, possibly due to it being contained in the mini-chromosome. Computational analyses reveal that *APiasL3* is part of a larger protein family which has members across blast fungus lineages, including the previously identified *AVR-Pias*. I investigated sequence conservation, identifying two highly-conserved pairs of cysteine residues shared by all members of the APiasL family, whilst an additional pair is present only in APiasL3 and APiasL4. I also performed selection analysis with several approaches, which returned a variety of residues which may be under positive or negative selection. Care in the

interpretation of these results is required, especially given the likelihood of false detection of positive selection and a history of misinterpretation as in the case of branch-site models. The 35A residue, however, was identified by each method as likely to be under positive selection, and as revealed by Figure 31, this residue is highly variable throughout the APiasL family. This variability, coupled with the previous indications of positive selection, may indicate that this residue has been selected for in APiasL3 due to a positive impact on pathogen survivability, whilst there may be a detrimental impact in other APiasL family members. The 35A residue is also one of three APiasL3 residues conserved with RanGAP2-WPP, which was highlighted using a structural similarity search method. This led to a hypothesis regarding a potential interaction interface coinciding with these residues, however, this work is highly dependent on how well the AlphaFold2 structural prediction for APiasL3 corresponds to the true structure, and the prediction used in this work was of only low to moderate confidence.

This work also described the identification of mini-chromosomes in the three Bangladeshi assemblies studied, using the same methods as previous work within the research group (Langner *et al.*, 2021) and as used in more recent work I contributed to on the Italian rice-infecting blast fungus population (Barragan *et al.*, 2024). A more developed understanding of the effector composition across the core genome and mini-chromosomes is important for further study of the adaptive mechanisms used by plant pathogens when faced with a challenging environment or host, such as in a new outbreak.

As mentioned in Chapter 1, previous work (Barragan *et al.*, 2024) identified a mini-chromosome horizontally transferred multiple times into clonal rice-infecting blast fungus, probably from *Eleusine*-infecting isolates. As discussed in that work, previous studies demonstrated that clonal isolates belonging to other fungal species can exchange mini-chromosomes via horizontal transfer under laboratory conditions, depending on the mechanism of parasexual recombination. Previous work (Zeigler *et al.*, 1997) identified signs of parasexual recombination in rice-infecting blast fungus field isolates, and it is possible that clonal blast fungus isolates are able to utilise this mechanism to increase genetic diversity. It is plausible that the distinctive presence and absence pattern of the mini-chromosome-borne *APiasL3* within the B71 lineage and throughout certain blast fungus lineages may be a result of horizontal transfer operating via this mechanism allowing clonal lineage diversification where sexual reproduction does not occur.

As previously discussed, in addition to the potential for horizontal transfer, the characteristics of mini-chromosomes allow pathogen adaptation and host range alteration through effector gain, duplication or function alteration (Croll and McDonald, 2012). Through taking advantage of a two-speed genome, blast fungus isolates carrying mini-chromosomes may utilise these fast-evolving genomic compartments to facilitate adaptation (Dong, Raffaele and Kamoun, 2015). Previous studies, such as (Gyawali *et al.*, 2023), have noted an increase in wheat-infecting isolates carrying mini-chromosomes which seemed to occur during the 1990s. That work used machine learning to predict an absence of mini-chromosomes in pre-1991 isolates and that all wheat-infecting isolates in their analysis which were collected after 2005 contained a mini-chromosome. This was not experimentally verified within that work. Whilst for many recently-collected wheat-infecting isolates it is not yet known through experimental means if they contain a mini-chromosome, experimental work shows that carrying a mini-chromosome is certainly not universal. There are links between isolate fertility and the prevalence of asexual reproduction with the increase in isolates carrying mini-chromosomes.

As shown by recent work (Latorre *et al.*, 2023), it is possible for sexual reproduction to occur between the recently arrived Zambian members of the B71 lineage and endemic isolates of blast fungus infecting on non-wheat hosts. Interactions of this type could account for a gain of *APiasL3* and *Art1_WB_ZM* in the Bangladeshi and Zambian wheat-infecting populations, respectively, if these effector candidates are present in local populations infecting on other hosts. As such, further investigation and comparison of the other blast fungus lineages present in Bangladesh and Zambia would be beneficial. A deeper understanding of if such events have occurred, or are likely to occur in future in both Zambia and Bangladesh, would both benefit understanding of the history of the effector candidates discussed in this work whilst also informing blast fungus management practices.

B71 lineage isolates may therefore be able to increase genetic diversity through interaction with members of blast fungus lineages infecting on local wild grasses, either through sexual reproduction, or through horizontal chromosome transfer. The former is limited by compatibility of mating types of the isolates in question, whilst the latter has been demonstrated to have occurred through mini-chromosome introduction into a clonal rice-infecting lineage multiple times through history.

Future directions

Investigations of genomic rearrangements are fundamentally limited by the availability of high quality genome assemblies. Areas of doubt in this work would likely be clarified by the addition of such information, such as if the integration of mini-chromosome sequence into the BTMP-S13-1 isolate represents the reality, and whether the current assembly is incorrectly missing a duplication of this sequence in the mini-chromosome. Improved genome assemblies would also increase the proportion of isolates, many with sequenced genomes but fragmented assemblies, which could be incorporated into the analysis, more fully representing the isolates present within the B71 lineage. As the separate outbreaks now present in South America, Asia and Africa develop on independent trajectories, mapping these changes over time will be particularly interesting.

As discussed in previous chapters, AlphaFold3 was released towards the end of the project, and greatly increased the confidence of structure prediction for Art1_WB_ZM. It is worthwhile to compare structures obtained with both AlphaFold3 and previous models, however, since the more recent version does show increased prediction confidence generally, in this case APiasL3 did not display improved prediction confidence.

In this work, the inability to generate *APiasL3*-deletion transformants hampered further experimental work, and it is plausible the presence of *APiasL3* within the mini-chromosome sequence led to being unable to completely remove *APiasL3* from the BTJP4-1 genome. Whilst it may be a time-consuming process, succeeding in generating such transformants would be valuable in determining any virulence impact resulting from the effector's presence.

Any future infection assays designed to investigate effectors present in the B71 lineage would benefit from using wheat cultivars that are grown in Zambia and Bangladesh. Incorporating wild grasses which grow there would also be beneficial due to the potential for cross-infection between crops and wild species. Detecting if the effector-deletion transformants are more or

less able to infect on locally-grown wild grasses than wheat would be valuable for assessing management strategies. The presence of *APiasL3* variants in rice-infecting blast fungus isolates makes certain rice cultivars another potential angle for investigation in such an experiment, to test if the wheat-infecting variant of APiasL3 is able to cause infection in rice, or vice versa. Additionally, the infection assays presented in this work did not attempt to measure how lesion size varied over time, nor to measure the final lesion size or quantity of spores produced, and it is possible that doing so might reveal less obvious impacts from the effector candidate deletion transformation process.

Identification of the target proteins of APiasL3 and Art1_WB_ZM would be important in identifying their function and role, which could be achieved by conducting a yeast two-hybrid screen. Additionally, attempting to remove the mini-chromosome from Bangladeshi and Zambian isolates, and performing infection assays in comparison with the wild type isolate would be an interesting approach to broadly identify an infection role for mini-chromosome-contained effectors. In members of the Bangladeshi population, this may have the side-effect of also removing *APiasL3*.

The pipeline I used to identify effector candidates is currently limited in being unable to deal with queries containing introns. Any further developments would be advised to investigate ways to handle these to broaden the range of possible query proteins that can be investigated. Additionally, use of DIAMOND instead of the currently-used BLAST would significantly increase search speed whilst having limited impacts on the sensitivity. If significantly larger datasets were to be used in future work, this would be a worthwhile improvement to the pipeline. Usage of more sophisticated selection analysis models than the branch model and branch-site model used in this work would be beneficial for increasing confidence in the residues highlighted to potentially be under positive selection. Further work using this pipeline could also be performed focussing on the effector diversity present within the clonal sub-lineages within the rice-infecting lineage.

With increasing accuracy of structural predictions and interactions between proteins through the use of tools such as AlphaFold, it may be beneficial to conduct a computational screen for likely interactors with APiasL3, perhaps focussing on NLR CC domains in wheat and Lolium, as well as wild grasses which grow in Bangladesh.

Determining the origin of the genomic regions identified in this study as present in Bangladeshi or Zambian isolates, but absent in B71, would be useful given that they contain effector candidates. Particularly, the segment in the Bangladeshi isolates containing the *APiasL3* candidate which is absent in other B71-lineage populations, is also present on the mini-chromosome. As described recently (Barragan *et al.*, 2024), investigation of similarity between the core genome and mini-chromosomes, both within and between isolates, can indicate differing evolutionary paths. Such techniques could be applied to the aligning and non-aligning regions used in this study to identify any segments with aberrant characteristics from the rest of the genome of that isolate. Machine learning techniques have been recently applied (Gyawali *et al.*, 2023) to predict mini-chromosome sequence in blast fungus based on short-read sequence data, however such approaches are highly dependent on volume and diversity of training data, which may not be available at this time. However, if a robust prediction tool is available, then it would be informative to screen the non-aligning genome segments used in this work, including those containing *Art1_WB_ZM* and *APiasL3*, in comparison with the other

portions of the Zambian and Bangladeshi isolate genomes. Alternatively, existing clustering techniques applied to the aligning and non-aligning genome segments used in this work might be similarly interesting.

Conclusion

During this project, I analysed Bangladeshi and Zambian wheat-infecting blast fungus genomes belonging to the B71 lineage, and identified one effector candidate, *APiasL3*, which is specific to the Bangladeshi population, and a second effector candidate, *Art1_WB_ZM*, which is specific to the Zambian population. I then attempted to generate effector-deletion transformants of both, and performed infection assays. The transformants in the Bangladeshi population were not successful, likely due to the effector presence on the mini-chromosome, and the Zambian transformant infection assay did not indicate any noticeable virulence effect from the loss of *Art1_WB_ZM*. Despite the lack of evidence for virulence impact, further experimentation and investigation is worthwhile, particularly given that *APiasL3* belongs to a broader family containing the known avirulence gene, *AVR-Pias*. It would also be worth repeating the infection assay experiment with the Zambian *Art1_WB_ZM* deletion transformant using wheat cultivars known to be grown in Zambia, since any virulence difference conferred by loss of the effector candidate may be visible in these cultivars but not in those used in the experiments in this work.

Whilst virulence changes associated with the loss of either population-specific effector candidate could not be identified in this work, in a clonal lineage which has spread onto multiple continents and continues to infect upon different host populations, it is worthwhile to further investigate these and other population-specific effectors due to their potential role in adaptation within this emerging and developing clonal pandemic lineage.

Appendix 1 – Code

BLAST heatmap pipeline

Genome .fasta files should be placed into a ‘genomes’ directory, and effectors/queries placed in an ‘effectors’ directory. The default version of the pipeline uses the tblastn option in terminal ~BLAST, but an alternative ‘step 1’ script is provided using the blastn option.

A file containing the names of the queries (usually effectors) in the desired order on the plot should be created, called ‘effectors_order.txt’. Each effector name should appear on a new line, and match the .fasta file input name (but with no ‘.fasta’ included in the contents of the effectors_list.txt file.

The same process should be followed for the genome .fasta files, but putting the genome names instead into a file named ‘genomes_order.txt’. It is recommended to use a simple command such as the following on the command line (assuming all the genomes you want to include are located in the ‘genomes’ directory, and that the ‘genomes’ directory is the working directory) to automate this process:

```
for genomes in *.fasta; do echo ${genomes%.fasta}; done >
../genomes_order.txt
```

Pipeline_Autoheatmap_basic.sh

The following code is the control script which calls the ‘step 1’ (BLAST) and ‘step 2’ (filtering) Bash scripts, followed by the Python plotting script.

```
#!/bin/bash
#This is the controller script for the basic, default pipeline using
python for heatmap plotting.

echo -e "Please enter the number corresponding to the option you wish
to select.\nWhich version of blast do you want to use?\n1:
tblastn\n2: blastn\n"
read blast_option
if (("$blast_option"=="1"))
then
     echo "running tblastn option"
     ./AutoHeatmap_pipeline_step_1_blast.sh
     wait
     ./AutoHeatmap_pipeline_step_2_filtering.sh
     wait
     python heatmap_default_seaborn.py > python_script_log.txt
elif (("$blast_option"=="2"))
then
                                          #Run the blastn version
of the pipeline
```

```
      echo "running blastn option"
         ./AutoHeatmap_pipeline_step_1_blast_variant_blastn.sh
      #blastn version
         wait
         ./AutoHeatmap_pipeline_step_2_filtering.sh
         wait
         python heatmap_default_seaborn.py
else
      echo "Not a valid option. Quitting. Please try again."
fi
```

AutoHeatmap_pipeline_step_1_blast.sh

```
#!/bin/bash
#Autoheatmap pipeline step 1: blast searches
#This step will conduct blast searches with the fasta files in the
genomes directory as the databases, and the fasta files in
#the effectors directory as the query sequences. Output will be saved
in the blast_output directory.

#Script by Angus Malmgren, TSL

echo "Beginning blast search step:"

GENOMES_DIR=genomes
EFFECTORS_DIR=effectors

#fasta files present in genome and effector directories:
GENOMES_FOUND=$GENOMES_DIR/*.fasta
EFFECTORS_FOUND=$EFFECTORS_DIR/*.fasta

#Loop through fasta files in genomes folder, and create blast
databases:
for fasta in $GENOMES_FOUND
do
    #Strip the preceeding directory name for database title naming
clarity:
    fasta_stripped=${fasta/#${GENOMES_DIR}\//}
    echo "Processing ${fasta} - generating blast database"
    makeblastdb -in $fasta -dbtype nucl -parse_seqids -title
"${fasta_stripped%.fasta}"
done

echo; echo "Database creation finished."

#Wait until all previous operations have finished:
wait

echo ""
```

```
echo "Starting blast searches:"
outer_counter=1

#Loop over the genomes in genomes folder, and effectors in effectors
folder, then perform blast searches with each combination:
for fasta2 in $GENOMES_FOUND
do
    echo "Processing genome: $outer_counter"
        inner_counter=1
    #Strip the preceding directory name for file naming:
    fasta2_stripped=${fasta2/#${GENOMES_DIR}\//}
    for effector in $EFFECTORS_FOUND; do
        echo "Processing query: $inner_counter"
        effector_stripped=${effector/#${EFFECTORS_DIR}\//}
        echo "Blast search using ${fasta2_stripped} database and
${effector_stripped} query"
            tblastn -db $fasta2 -query $effector -out
blast_output/blast_out_db_${fasta2_stripped%.fasta}_query_${effector_
stripped%.fasta}.out -seg no -comp_based_stats 0 -outfmt '6 qseqid
qlen sseqid sstart send length pident qcovs evalue sseq'
        let "inner_counter+=1"
        echo ""
    done
    let "outer_counter+=1"
    echo ""
done

echo; echo "Ending blast search step"
exit
```

AutoHeatmap_pipeline_step_1_blast_variant_blastn.sh

```
#!/bin/bash
#Autoheatmap pipeline step 1: blast searches
#This step will conduct blast searches with the fasta files in the
genomes directory as the databases, and the fasta files in
#the effectors directory as the query sequences. Output will be saved
in the blast_output directory.

#Note this is almost exactly the same script as
AutoHeatmap_pipeline_step_1_blast.sh, except this version uses blastn
(nucleotide input) whereas the other version is the original, using
tblastn
#Note that the regions you want to analyse should still go in the
"effectors" folder. Also, note that in the tblastn analysis I
eventually disabled "seg" and "comp_based_stats".

#Script by Angus Malmgren, TSL
```

```
echo "Beginning blast search step:"

GENOMES_DIR=genomes
EFFECTORS_DIR=effectors

#fasta files present in genome and effector directories:
GENOMES_FOUND=$GENOMES_DIR/*.fasta
EFFECTORS_FOUND=$EFFECTORS_DIR/*.fasta

#Loop through fasta files in genomes folder, and create blast
databases:
for fasta in $GENOMES_FOUND
do
    #Strip the preceding directory name for database title naming
clarity:
    fasta_stripped=${fasta/#${GENOMES_DIR}\//}
    echo "Processing ${fasta} - generating blast database"
    makeblastdb -in $fasta -dbtype nucl -parse_seqids -title
"${fasta_stripped%.fasta}"
done

echo; echo "Database creation finished."

#Wait until all previous operations have finished:
wait

echo ""
echo "Starting blast searches:"
outer_counter=1

#Loop over the genomes in genomes folder, and effectors in effectors
folder, then perform blast searches with each combination:
for fasta2 in $GENOMES_FOUND
do
    echo "Processing genome: $outer_counter"
        inner_counter=1
    #Strip the preceding directory name for file naming:
    fasta2_stripped=${fasta2/#${GENOMES_DIR}\//}
    for effector in $EFFECTORS_FOUND; do
        echo "Processing query: $inner_counter"
        effector_stripped=${effector/#${EFFECTORS_DIR}\//}
        echo "Blast search using ${fasta2_stripped} database and
${effector_stripped} query"
            #blastn -db $fasta2 -query $effector -out
blast_output/blast_out_db_${fasta2_stripped%.fasta}_query_${effector_
stripped%.fasta}.out -seg no -comp_based_stats 0 -outfmt '6 qseqid
qlen sseqid sstart send length pident qcovs evalue sseq'
```

```
        blastn -db $fasta2 -query $effector -out
blast_output/blast_out_db_${fasta2_stripped%.fasta}_query_${effector_
stripped%.fasta}.out -outfmt '6 qseqid qlen sseqid sstart send length
pident qcovs evalue sseq'
        let "inner_counter+=1"
        echo ""
    done
    let "outer_counter+=1"
    echo ""
done

echo; echo "Ending blast search step"
exit
```

AutoHeatmap_pipeline_step_2_filtering.sh

```
#!/bin/bash
#Autoheatmap pipeline step 2: filtering output from blast searches
#This step in the pipeline is for filtering and organising the output
of the blast searches.

#Script by Angus Malmgren

echo "Beginning blast search filtering step:"

BLAST_OUT_DIR=blast_output
TOP_BLAST_HITS_DIR=top_blast_hit_results

BLAST_OUT_FOUND=$BLAST_OUT_DIR/*.out

COVERAGE_VAL=0.9
P_IDENT_THRESHOLD_1=50
#P_IDENT_THRESHOLD_2=50

for blast_out in $BLAST_OUT_FOUND
do
    blast_out_stripped=${blast_out/#${BLAST_OUT_DIR}\//}
    echo "Processing ${blast_out_stripped}"
    blast_out_stripped_short=${blast_out_stripped%.out}

    awk -v name="$blast_out_stripped_short" '{ print
$1,$2,$3,$4,$5,$6,$7,$8,$9,$10,name}' ${blast_out} >
filtering_output/${blast_out_stripped%.out}_all_output.out
done

#Collect information into format for plotting
HEATMAP_INFO_DIR=heatmap_info
#TOP_HITS_FOUND=${TOP_BLAST_HITS_DIR}/*.out
```

```
ALL_HITS=filtering_output/*_all_output.out

echo "perc_ident comparison seq q_len len sseqid sstart send evalue"
> ${HEATMAP_INFO_DIR}/heatmap_perc_ident_data_all_hits.txt

echo "Adding % identity from top blast hits to heatmap data file:"

#Create a fasta file in which to store the genomic sequences for the
best blast hits for each effector:
touch best_hit_genomic_sequences.fasta

for hit in $ALL_HITS
do
    hit_stripped=${hit/#filtering_output\//}
    echo "Adding heatmap data from ${hit_stripped}"
    blast_name_temp=${hit_stripped/#blast_out_/}
    blast_name=${blast_name_temp%.out}
    awk -v name="$blast_name" '{print $7,name,$10,$2,$6,$3,$4,$5,$9}'
${hit} >> ${HEATMAP_INFO_DIR}/heatmap_perc_ident_data_all_hits.txt

    #Add sequence to best_hit_genomic_sequences.fasta:
    #awk -F "\t" -v name="$blast_name" '{print
">comparison="name}"\n"$10' ${hit} >>
best_hit_genomic_sequences.fasta
#   awk -F "\t" -v name="$blast_name" '{print
">comparison="name"\n"$10}' ${hit} >>
best_hit_genomic_sequences.fasta
done

echo; echo "Ending blast search filtering step"
exit
```

heatmap_default_seaborn.py

```
#!/usr/bin/env python3

###################
#import libraries:#
###################

# Plotting functionality:
import matplotlib.pyplot as plt
# Data handling
import pandas as pd
# Mathematical functions:
import numpy as np
# Plotting aesthetics and functionality:
import seaborn as sns
```

```
# System functions:
import sys
# Tree data handling:
from ete3 import Tree
# Sequence handling functionality:
from Bio import SeqIO
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord

##################
#Define functions:

def plot_heatmap_figure(df, name, plotting_color_midpoint, vminimum,
vmaximum):
    '''
    A function to plot heatmaps, taking as input the output name, a
dataframe containing plotting data and a numerical value
    or string instructing where the colour scale midpoint should be.
Also, takes in minimum and maximum values of colour scale.
    '''

    f, ax = plt.subplots(figsize=(14.0, 60.0)) # Define figure size
    mask_nan=df.isnull() #create a boolean array indicating whether
entries are NaN (Not a Number) so they can be mask out in plotting

    # Plot data using automatic heatmap colour midpoint selection:
    if plotting_color_midpoint == 'auto':
        sns.heatmap(df,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, cbar_kws={"shrink": 0.3, "label": "%
identity"},ax=ax, fmt=".1f",annot_kws={'size':8},square=True,
mask=mask_nan)
    # Plot data using manually chosen heatmap colour midpoint
selection:
    else:
        sns.heatmap(df,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, center=plotting_color_midpoint,
cbar_kws={"shrink": 0.3, "label": "% identity"},ax=ax,
fmt=".1f",annot_kws={'size':8},square=True, mask=mask_nan)

    plt.savefig(name) #Save the plot - alter the .png for a different
file format
    return


def plot_double_heatmap_figure(df1, df2, name,
plotting_color_midpoint, vminimum, vmaximum):
    '''
```

```
    A function (specialised version of the above function
plot_heatmap_figure) to plot two heatmaps side-by-side, corresponding
to the top and 2nd best blast hit matches.
    Taking as input the output name, two dataframes containing
plotting data and a numerical value
    or string instructing where the colour scale midpoint should be.
Also, takes in minimum and maximum values of colour scale.
    '''

    left_cbar_draw=False #Draw colourbar on left plot

    f, (ax1,ax2) = plt.subplots(1,2,figsize=(40.0, 60.0),
sharey=True)
    mask_nan_1=df1.isnull() #create boolean array indicating if
entries are NaN (Not a Number) to allow masking
    mask_nan_2=df2.isnull()

    # Plot data using automatic heatmap colour midpoint selection:
    if plotting_color_midpoint == 'auto':
        sns.heatmap(df1,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, cbar_kws={"shrink": 0.3, "label": "%
identity"}, fmt=".1f",annot_kws={'size':8},square=True,
mask=mask_nan_1, ax=ax1, cbar=left_cbar_draw)
        ax1.set_title("Best hit")
        sns.heatmap(df2,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, cbar_kws={"shrink": 0.3, "label": "%
identity"}, fmt=".1f",annot_kws={'size':8},square=True,
mask=mask_nan_2, ax=ax2)
        ax2.set_title("Second best hit")
    # Plot data using manually chosen heatmap colour midpoint
selection:
    else:
        sns.heatmap(df1,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, center=plotting_color_midpoint,
cbar_kws={"shrink": 0.3, "label": "% identity"},
fmt=".1f",annot_kws={'size':8},square=True, mask=mask_nan_1, ax=ax1,
cbar=left_cbar_draw)
        ax1.set_title("Best hit")
        sns.heatmap(df2,annot=True, linewidths=.5, cmap='RdYlBu_r',
vmin=vminimum, vmax=vmaximum, center=plotting_color_midpoint,
cbar_kws={"shrink": 0.3, "label": "% identity"},
fmt=".1f",annot_kws={'size':8},square=True, mask=mask_nan_2, ax=ax2)
        ax2.set_title("Second best hit")

    f.tight_layout()
    #plt.subplots_adjust(wspace=0, hspace=0)
    plt.savefig(name) #Save the plot - alter the .png for a different
file format
```

```
    return


def plot_heatmap_figure_clustered_queries(df1, name,
plotting_color_midpoint, vminimum, vmaximum):
    '''
    A function (specialised version of the above function
plot_double_heatmap_figure) to plot one heatmap, corresponding to the
best blast hit match, with an x-axis (queries) clustered with a
dendrogram.
    Taking as input the output name, one dataframe containing
plotting data and a numerical value
    or string instructing where the colour scale midpoint should be.
Also, takes in minimum and maximum values of colour scale.
    '''

    left_cbar_draw=False #Draw colourbar on left plot

    f, ax1 = plt.subplots(figsize=(300.0, 240.0))
    mask_nan_1=df1.isnull() #create a boolean array indicating
whether entries are NaN (Not a Number) so they can be mask out in
plotting

    fig_width=120.0
    fig_height=60.0

    cluster_df1=df1.fillna(0.0000000001)

    # Plot data using automatic heatmap colour midpoint selection:
    if plotting_color_midpoint == 'auto':
        sns.clustermap(cluster_df1,annot=True, linewidths=.5,
cmap='RdYlBu_r', vmin=vminimum, vmax=vmaximum,
fmt=".1f",annot_kws={'size':8},square=True, mask=mask_nan_1,
row_cluster=False, figsize=(fig_width, fig_height))#,
cbar=left_cbar_draw
        ax1.set_title("Best hit")
    # Plot data using manually chosen heatmap colour midpoint
selection:
    else:
        sns.clustermap(cluster_df1,annot=True, linewidths=.5,
cmap='RdYlBu_r', vmin=vminimum, vmax=vmaximum,
center=plotting_color_midpoint,
fmt=".1f",annot_kws={'size':8},square=True, mask=mask_nan_1,
cbar=left_cbar_draw, row_cluster=False, figsize=(fig_width,
fig_height))
        ax1.set_title("Best hit")

    f.tight_layout()
```

```
    plt.savefig(name) #Save the plot - alter the .png for a different
file format
    return


def output_fasta_from_df(df, name, long_description=True):
    '''
    A function that takes as input the output file name and the
dataframe containing filtered blast hit results, one hit per genome-
protein comparison.
    '''
    print()
    print()
    print('Outputting fasta file: ', name)
    output_fasta_records_genome=df['genome'].tolist()
    output_fasta_records_effector=df['effector'].tolist()
    output_fasta_records_seq=df['seq'].tolist()

    print('Number of genome entries:',
len(output_fasta_records_genome))
    print('Number of effector entries:',
len(output_fasta_records_effector))
    print('Number of sequence entries:',
len(output_fasta_records_seq))
    print('These three numbers should be identical')

    if
(len(output_fasta_records_genome)!=len(output_fasta_records_effector)
) or
(len(output_fasta_records_seq)!=len(output_fasta_records_effector)):
        print('ERROR: lists of values read from dataframe columns are
not equal in length.')
    print()

    #element-wise, combine genome and effector names to get list
containing sequence ids:
    output_fasta_records_id_list=df[['genome','effector']].agg('_'.jo
in,axis=1).tolist()

    #Choose whether to use a long or short form of the description
field (long form includes % identity and coverage):
    if long_description==True:
        output_fasta_records_list=[]
        for index in range(len(output_fasta_records_genome)):
            description_string="Genome={} Protein={}
algn_len_over_q_len={:.3f} perc_ident={:.2f} Genome_contig={}
Genome_start_coord={}
Genome_end_coord={}".format(df.iloc[index,8],df.iloc[index,9],df.iloc
```

```
[index,10],df.iloc[index,0],df.iloc[index,4],df.iloc[index,5],df.iloc
[index,6])

            #Create a Biopython SeqRecord:
            record=SeqRecord(Seq(df.iloc[index,1]),
id=output_fasta_records_id_list[index],
description=description_string)

            output_fasta_records_list.append(record) #Add each record
to list

    elif long_description==False:
        output_fasta_records_list=[]
        for index in range(len(output_fasta_records_genome)):
            description_string="Genome={}
Protein={}".format(df.iloc[index,8],df.iloc[index,9])

            #Create a Biopython SeqRecord:
            record=SeqRecord(Seq(df.iloc[index,1]),
id=output_fasta_records_id_list[index],
description=description_string)

            output_fasta_records_list.append(record) #Add each record
to list

    #Using Biopython, create a fasta file containing the sequences
collected in the list:
    SeqIO.write(output_fasta_records_list, name, "fasta")
    Return


def order_data_by_tree(df, tree_file):
    print()
    #Get genome order from tree file:
    t=Tree(tree_file)

    print('Tree file name ordering:')
    tree_genome_order=t.get_leaf_names()

    #Replace all '.' with '_' to match string format used by kSNP3...
    print(df.index)
    genome_names_old=df.index
    genome_names_updated=genome_names_old.str.replace('.','_',regex=F
alse)
    df=df.reindex(genome_names_updated)
    #Reindex dataframe according to tree genome ordering:
    df=df.reindex(tree_genome_order)
    print(df)
```

```
    return df

def remove_specific_sequences(df_all_data, fasta_with_seq_to_remove,
ignore_hyphens=False, drop_all_duplicate_seq=False):
    '''
    This function will remove all sequences identical to entries in
fasta_with_seq_to_remove, from df_all_data. This should be
    performed on the dataframe during the filtering stage.

    if ignore_hyphens is False, then only if two sequences are
exactly equal will the hit be filtered out. If ignore_hyphens is
True,
    then if the hit sequence (coming from the genome) contains
hyphens, but is otherwise exactly equal to a sequence in the file
    fasta_with_seq_to_remove, it will be removed.
    '''

    print()
    print()
    print('Removing sequences present in file: ',
fasta_with_seq_to_remove)
    print('Number of entries in dataframe before removing found
sequences:', len(df_all_data))
    seq_list=[]
    #Read in the fasta file:
    for record in SeqIO.parse(fasta_with_seq_to_remove, "fasta"):
        print(record.seq)
        seq_list.append(record.seq)#[0])
    print()

    #find hits which are identical, but also if they have '-'
characters different:
    if ignore_hyphens == True:
        #Create a new column to then strip out all '-' characters in
the sequence:
        df_all_data['seq2']=df_all_data['seq']

        #remove '-' character:
        df_all_data['seq2']=df_all_data['seq2'].str.replace('-','')
        #perform filtering using the new column
        df_removed_data=df_all_data[df_all_data['seq2'].isin(seq_list
)]
        #Remove rows in dataframe if they contain one of these
sequences in the list:
        df_all_data=df_all_data[~df_all_data['seq2'].isin(seq_list)]
```

```
        #Remove entries which have duplicated sequences, to get only
unique entries. THIS APPLIES TO THE ENTIRE DATASET AND WILL REMOVE
ANY DUPLICATED SEQUENCE
        if drop_all_duplicate_seq == True:
            df_all_data=df_all_data.drop_duplicates(subset=['seq2'])

        #Drop the new column:
        df_all_data=df_all_data.drop(columns=['seq2'],axis=1)

    #if ignore_hyphens is false, use the original column instead -
this will find only hits with exactly the same sequence:
    elif ignore_hyphens == False:
        df_removed_data=df_all_data[df_all_data['seq'].isin(seq_list)
]
        #Remove rows in dataframe if they contain one of these
sequences in the list:
        df_all_data=df_all_data[~df_all_data['seq'].isin(seq_list)]
#115 seq -> 99 sequences

        #Remove entries which have duplicated sequences, to get only
unique entries.
        df_all_data=df_all_data.drop_duplicates(subset=['seq'])


    print('Number of entries in dataframe after removing found
sequences:', len(df_all_data))

    output_fasta_from_df(df_all_data,
"sequences_unfiltered_all_hits_with_sequences_removed.fa")
    output_fasta_from_df(df_removed_data,
"sequences_unfiltered_removed_hits_with_sequences_removed.fa")
    print()
    print()
    return df_all_data

def return_nonredundant_seq(df_all_data, output_name,
ignore_hyphens=False):
    '''
    This function will return all unique sequences in the sequence
data given, and output to a file. It is useful for identifying
    the non-redundant hits returned by the blast search pipeline.
This can then be fed into a second run  of the pipeline, with high
    stringency coverage (e.g. 100%) and % identity (e.g. 99%)
thresholds, to show the presence-absence of variants across lineages.

    if ignore_hyphens is False, then only if two sequences are
exactly equal (including any gaps) will the hit be identified as
```

```
    non-unique (one of the sequences will be removed). i.e. if False,
two identical sequences, but one having an additional gap ('-'),
    will be counted as two unique sequences and output to the file.
If ignore_hyphens is True, then if the hit sequence (coming from
    the genome) contains hyphens, but is otherwise exactly equal to
another sequence, one of the sequence entries will be removed.
    '''
    print()
    print("non-redundant hit filtering, input data:")
    print(df_all_data)
    print()
    #find hits which are unique, but also removes and '-' character
(so finds hits which are unique, unless there is a gap/insertion):
    if ignore_hyphens == True:
        #Create a new column to then strip out all '-' characters in
the sequence:
        df_all_data['seq2']=df_all_data['seq']
        #remove '-' character:
        df_all_data['seq2']=df_all_data['seq2'].str.replace('-','')

        #Remove entries which have duplicated sequences, to get only
unique entries. THIS APPLIES TO THE ENTIRE DATASET AND WILL REMOVE
ANY DUPLICATED SEQUENCE
        df_all_data=df_all_data.drop_duplicates(subset=['seq2'])
        #Drop the new column:
        df_all_data=df_all_data.drop(columns=['seq2'],axis=1)

    #if ignore_hyphens is false, use the original column instead -
this will find only hits that are unique, even if that uniqueness is
gue to a gap/insertion:
    elif ignore_hyphens == False:
        #Remove entries which have duplicated sequences, to get only
unique entries.
        df_all_data=df_all_data.drop_duplicates(subset=['seq'])

    print('Number of entries in dataframe after removing found
sequences:', len(df_all_data))

    output_fasta_from_df(df_all_data, output_name)
    print()

    return

print("Beginning Python blast search filtering and heatmap plotting
script...")
print("")

###########
```

```
#Define parameters:
############

#1st run: (find groups/homologs amongst lineages)
      # % identity = 40%
      # % coverage = 60%
#2nd run: (high-resolution presence/absence of homologs)
      # % identity = 90%
      # % coverage = 90%
heatmap_masking_threshold = 90.0 # Define the % identity threshold
below which a hit is not displayed in the heatmap - it will be a NaN.
Set between 0.0-100.0
min_coverage_threshold = 0.90 # Set the threshold for coverage
(alignment length / query length) below which blast hits will be
filtered out (0.0-1.0).

#Comment out one of the following 2 options depending on how you wish
to colour the heatmap:
plotting_color_midpoint = 50.0 # Define the % identity value at which
the midpoint of the heatmap colour change will be (does not need to
be at the centre)
                                                        # Express as a
float within the range of the data being plotted in the heatmap -
e.g. 88.0
plotting_color_midpoint = 'auto' # Automatically set heatmap colour
midpoint.

heatmap_colour_scale_minimum= 90.0 # Set the minimum value for the
colourbar (for % identity, this should be between 0% - 100%)
heatmap_colour_scale_maximum= 100.0 # Set the maximum value for the
colourbar (for % identity, this should be between 0% - 100%)
############


##################
# Begin main code:

print("Using masking threshold (% identity) of
{:04.1f}%".format(heatmap_masking_threshold))
print("Using Minimum coverage threshold of
{:05.1f}%".format(min_coverage_threshold*100.0))

#Read in datafile as a pandas 'dataframe':
heatmap_df=pd.read_csv('heatmap_info/heatmap_perc_ident_data_all_hits
.txt', delimiter=' ', na_values='no_match')

#Read in the genomes_order.txt and effectors_order.txt files as
pandas dataframes:
```

```
genomes_order_df=pd.read_csv('genomes_order.txt', delimiter=' ',
names=['Genome'])
effectors_order_df=pd.read_csv('effectors_order.txt', delimiter=' ',
names=['Effector'])
print('Found effectors: ', effectors_order_df)

#Initialise two empty lists to store values in:
genomes_column=[] #Initialise a new list to serve as the genome
column in the heatmap_df
effector_column=[] #Initialise a new list to serve as the effector
column in the heatmap_df

#Clean characters that cause issues with display of heatmap data:
heatmap_comparison_old=heatmap_df['comparison'] #Take the
'comparison' column of the heatmap and assign it to
heatmap_comparison_old
heatmap_comparison=heatmap_comparison_old.str.replace('.','_',regex=F
alse) #Replace any instances of '.' character in the 'comparison'
column with a '_'

print()
print('Heatmap_comparison: ', heatmap_comparison)


#Identify genome names present in the genomes_order.txt file:
for heatmap_entry in heatmap_comparison: #loop through entries in
heatmap_comparison
    #loop through genome names in the genomes_order.txt file:
    for genomes_order_entry_temp in genomes_order_df['Genome']:
        genomes_order_entry=genomes_order_entry_temp.replace('.','_')
#replace '.' characters with '_'

        #if genome name from genomes_order.txt is found within the
'comparison' column from heatmap_perc_ident_data_all_hits.txt, add
that genome name to the genomes_column list:
        if genomes_order_entry in heatmap_entry:
            genomes_column.append(genomes_order_entry)
    #loop through effector names in the effectors_order.txt file:
    for effectors_order_entry in effectors_order_df['Effector']:
        if effectors_order_entry in heatmap_entry:
            effector_column.append(effectors_order_entry)

#Create a 'genome' column in heatmap_df to store the name of which
genome was used in the comparison
heatmap_df['genome']=genomes_column

#Create an 'effector' column in heatmap_df to store the name of which
effector was used in the comparison
```

```
heatmap_df['effector']=effector_column

#Remove the column 'comparison' from heatmap_df to reduce clutter:
heatmap_df=heatmap_df.drop(columns=['comparison'])

#Write a machine-readable version of the data:
heatmap_df.to_csv('heatmap_values_machine_readable.csv')


############
#Filtering:

print()
print('Filtering data...')

#Separate pseudogenes into separate file:
pseudogenes_df=heatmap_df[heatmap_df.seq.str.contains('\*')] # Create
new dataframe which contains only entries containing the * character
print()
print('Found ', len(pseudogenes_df), ' pseudogenes (containing *
character)')
print('pseudogene emtries:\n', pseudogenes_df)
print()

#Remove all entries in main dataframe which contain the * character:
print('Number of entries in data before pseudogenes removed: ',
len(heatmap_df))
heatmap_df=heatmap_df[~heatmap_df.seq.str.contains("\*")]
print('Number of entries in data after pseudogenes removed: ',
len(heatmap_df))


#Get entries without Methionine in start position:
no_methionine_start_df=heatmap_df[~heatmap_df.seq.str.startswith('M')
]
print('Found ', len(no_methionine_start_df), ' sequences without
Methionine at beginning.')

print()
print('Dataset before filtering (but after removing pseudogenes):')

#Create column containing alignment length/query length:
heatmap_df['algn_len_over_q_len']=heatmap_df['len']/heatmap_df['q_len
']
print(heatmap_df)
print()
print('Dataset after applying filtering:')
```

```
# As a basic filtering step, select the entries which have q_len and
len matching to a certain %, then take only the blast hits which have
# % identity greater than the threshold:
heatmap_df=heatmap_df.loc[(heatmap_df['perc_ident']>=heatmap_masking_
threshold)&(heatmap_df['algn_len_over_q_len']>=min_coverage_threshold
)] #remove hits from the dataframe if they do not meet

      # minimum % identity requirements and minimum coverage
requirements - both specified at top of script.
print()
print("Heatmap df:")
print(heatmap_df)
print()

# The following block can be used to remove specific sequences
(remove the backticks surrounding the block)
'''
#Remove specific sequences:
heatmap_df=remove_specific_sequences(df_all_data=heatmap_df,
fasta_with_seq_to_remove="PWL_Family_non-redundant_rmPseudo.fasta",
ignore_hyphens=True, drop_all_duplicate_seq=False)
'''
print()
print('Dataset after applying filtering and taking only the top hit
(by % identity) for each genome-effector comparison:')
# Take only the highest % identity hit for each comparison (assuming
it is greater than the % identity plotting threshold):
    # Get the top hits for each genome-protein/effector comparison
using groupby and idxmax.
heatmap_df_best_perc_ident=heatmap_df.loc[heatmap_df.groupby(by=['gen
ome','effector'])['perc_ident'].idxmax()]
print('Best hits in heatmap by % identity:')
print(heatmap_df_best_perc_ident)
print()

#To get second best hits, remove top hits from dataframe and repeat
groupby -> idxmax method:
temporary_heatmap_df=heatmap_df.drop(index=heatmap_df.groupby(by=['ge
nome','effector'])['perc_ident'].idxmax(), inplace=False)
heatmap_df_second_best_perc_ident=temporary_heatmap_df.loc[temporary_
heatmap_df.groupby(by=['genome','effector'])['perc_ident'].idxmax()]
print('Second best hits in heatmap by % identity:')
print(heatmap_df_second_best_perc_ident)

#Get entries without Methionine in start position from top hits:
no_methionine_start_heatmap_df_best_perc_ident=heatmap_df_best_perc_i
dent[~heatmap_df_best_perc_ident.seq.str.startswith('M')]
```

```
print()
print('Found ', len(no_methionine_start_heatmap_df_best_perc_ident),
' sequences without Methionine at beginning from the best hits
data.')
print('These are the entries:')
print(no_methionine_start_heatmap_df_best_perc_ident)
print()

print()
print('Outputting fasta files...')

# Save sequences of best hits to separate fasta file: (doesn't
include pseudogenes)
output_fasta_from_df(heatmap_df_best_perc_ident,
"sequences_filtered_top_perc_ident.fa")

# Save sequences of second best hits to a separate fasta file:
(doesn't include pseudogenes)
output_fasta_from_df(heatmap_df_second_best_perc_ident,
"sequences_filtered_second_best_perc_ident.fa")

# Save pseudogene sequences in a separate fasta file:
output_fasta_from_df(pseudogenes_df,
"sequences_unfiltered_pseudogenes.fa", long_description=False)

#Output nonredundant (unique) protein hits:
return_nonredundant_seq(df_all_data=heatmap_df_best_perc_ident,
ignore_hyphens=False,
output_name="sequences_filtered_non_redundant_top_hits.fa")
return_nonredundant_seq(df_all_data=heatmap_df_second_best_perc_ident
, ignore_hyphens=False,
output_name="sequences_filtered_non_redundant_second_best_hits.fa")

##########
#Pivot the dataframe to change structure:
print()
print('Pivoting dataframe...')
heatmap_df_best_perc_ident=heatmap_df_best_perc_ident.pivot(index='ge
nome', columns='effector', values='perc_ident')

heatmap_df_second_best_perc_ident=heatmap_df_second_best_perc_ident.p
ivot(index='genome', columns='effector', values='perc_ident')

print()
print('Reordering data using tree files...')
heatmap_df_best_perc_ident=order_data_by_tree(df=heatmap_df_best_perc
_ident,tree_file="Tree/outtree.resolved.tree.parsimony.rerooted.tree"
)
```

```
heatmap_df_second_best_perc_ident=order_data_by_tree(df=heatmap_df_se
cond_best_perc_ident,tree_file="Tree/outtree.resolved.tree.parsimony.
rerooted.tree")

#Note: if plotting 1st and 2nd best hits on the same plot, make sure
the reordering is done using the same file for both 1st and 2nd best
datasets!

#Use seaborn for heatmap generation:
print()
print('Plotting data...')
sns.set_theme() #Use seaborn colour schemes and styles for plotting
plot_heatmap_figure(df=heatmap_df_best_perc_ident,
name='Autoheatmap_plot_seaborn_heatmap_best_match.png',
plotting_color_midpoint=plotting_color_midpoint,
vminimum=heatmap_colour_scale_minimum,
vmaximum=heatmap_colour_scale_maximum)
plot_heatmap_figure(df=heatmap_df_second_best_perc_ident,
name='Autoheatmap_plot_seaborn_heatmap_second_best_match.png',
plotting_color_midpoint=plotting_color_midpoint,
vminimum=heatmap_colour_scale_minimum,
vmaximum=heatmap_colour_scale_maximum)
plot_double_heatmap_figure(df1=heatmap_df_best_perc_ident,df2=heatmap
_df_second_best_perc_ident,
name='Autoheatmap_plot_seaborn_heatmap_joint_first_and_second_best_ma
tch.png', plotting_color_midpoint=plotting_color_midpoint,
vminimum=heatmap_colour_scale_minimum,
vmaximum=heatmap_colour_scale_maximum)

plot_heatmap_figure_clustered_queries(df1=heatmap_df_best_perc_ident,
name='Autoheatmap_plot_seaborn_clustered_queries_heatmap_joint_first_
and_second_best_match.png',
plotting_color_midpoint=plotting_color_midpoint,
vminimum=heatmap_colour_scale_minimum,
vmaximum=heatmap_colour_scale_maximum)

heatmap_df.to_csv('heatmap_values_human_readable.csv')

print()
print('Ending python script')
```

Nucmer alignment

nucmer_alignment_batch_call.sh

```
#!/bin/bash

#Script for nucmer alignment:
#By Angus Malmgren, TSL

echo "Beginning nucmer alignments:"

#The directory containing the ref genome:
REF_GENOME_DIR=$1
#The genome name: include .fasta
REF_GENOME=$2
QRY_GENOMES_DIR=$3
FILTER_ALIGNMENTS_SHORTER_THAN=$4

REF_GENOME_FULL_PATH=${REF_GENOME}${REF_GENOME_DIR}

for QRYS in $QRY_GENOMES_DIR/*.fasta; do
     QRYS=${QRYS/#${QRY_GENOMES_DIR}\//}
     echo ${QRYS%.fasta}
     nucmer $REF_GENOME_DIR/$REF_GENOME $QRY_GENOMES_DIR/$QRYS --
prefix "nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}"
     wait
     delta-filter -l ${FILTER_ALIGNMENTS_SHORTER_THAN}
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}.delta >
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b.delta
     wait
     show-coords -r -c -l
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b.delta >
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b.coords
     wait
     #Produce dotplots: make one that is reordered+oriented to
cluster largest hits near main diagonal, and produce another dotplot
which does not reorder contigs
     #use -l option: Layout a multiplot by ordering and orienting
sequences such that the largest hits cluster near the main diagonal
     mummerplot -l --png --color -p
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b_reordered_contigs_mummerplot
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b.delta
     wait
```

```
     mummerplot --png --color -p
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b_mummerplot
nucmer_${REF_GENOME%.fasta}_${QRYS%.fasta}_filtered_${FILTER_ALIGNMEN
TS_SHORTER_THAN}b.delta
done
```

nucmer_alignment_batch_call_part_2_plotting.sh

```
#!/bin/bash

echo "Beginning nucmer alignments plotting step:"

#Call this script once you have called
nucmer_alignment_batch_call.sh, and edited the .gp files to get it to
work (comment out mouse clipboardformat line)

#The directory containing the alignments to plot:
DIR=$1

for ALGN in $DIR/*.gp; do
     ALGN=${ALGN/#${DIR}\//}
     echo ${ALGN%.gp}

     #Use Here Documents to get gnuplot commands to execute:
     gnuplot <<-MARKER
     set terminal png
     set lmargin 20
     load "${ALGN}"
     MARKER
done
```

Extracting non-aligning genomic regions

extract_nonaligning_regions_control_script.sh

```
#!/bin/bash

#This bash script controls the pipeline to align genomes to a
reference with nucmer, and then extract the non-aligning and aligning
regions of those genomes.

#Call this script by using: bash
extract_nonaligning_regions_control_script.sh <GENOME FASTA FILE>
#This script should be run from inside the main directory, which
should only contain the two following directories - one containing
```

```
.fasta files for the query genomes, one containing a single reference
.fasta file

#Script by Angus Malmgren, TSL

main_DIR=$(pwd)

REF_genomes_DIR=REF_genome
QRY_genomes_DIR=QUERY_genomes

REFERENCE_genome=$1

#Make a directory to contain the extracted regions:
mkdir Extracted_nonaligning_regions
Extracted_regions_DIR=Extracted_nonaligning_regions

#Make a directory to contain the translated sequences:
mkdir translated_sequences

#Give the nucmer alignment filtering threshold (for removing
alignments shorter than this value) in bases:
Filtering_threshold=$2

#Create a directory to store the alignments in:
mkdir nucmer_alignments
cd nucmer_alignments

echo "-"
echo "-"
echo "-"

#Do alignments:
     #Only give one reference file in reference directory
bash
~/Documents/Projects/UtilityScripts/Easy_nucmer_alignments/nucmer_ali
gnment_batch_call.sh ${main_DIR}/${REF_genomes_DIR}
${REFERENCE_genome} ${main_DIR}/${QRY_genomes_DIR}
${Filtering_threshold}
cd ..

echo "-"
echo "-"
echo "-"

#Extract non-aligning regions, for each genome-reference comparison
cd nucmer_alignments
for Genome_QRY in ${main_DIR}/${QRY_genomes_DIR}/*.fasta
do
```

```
	echo "Analysing query genome:"
	echo ${Genome_QRY}
	COORDS_file_temp2=${Genome_QRY%.fasta}
	echo ${COORDS_file_temp2}
	COORDS_file_temp3=${COORDS_file_temp2/#${main_DIR}\//}
	COORDS_file_temp=${COORDS_file_temp3/#${QRY_genomes_DIR}\//}
	echo "COORDS file"
	echo ${COORDS_file_temp}
	COORDS_file=nucmer_${REFERENCE_genome%.fasta}_${COORDS_file_tem
p}_filtered_${Filtering_threshold}b.coords
	echo "Extracting regions of query genome which align and do not
align with the reference genome, under given alignment length
threshold"
	bash
~/Documents/Projects/UtilityScripts/non_aligning_sequence_extractor/e
xtract_nonaligning_regions_QRY.sh ${COORDS_file} ${Genome_QRY}
${main_DIR}
	echo "--"
	echo "--"
done

cd ../translated_sequences

#Loop through the non-aligning regions and use script from Joe Win to
translate and extract regions between a start and stop codon:
for nonaligning_region in
${main_DIR}/${Extracted_regions_DIR}/*_non_aligning_region.fasta
do
	Extract_region_path=${main_DIR}/${Extracted_regions_DIR}
	nonaligning_region_file_temp=${nonaligning_region%.fasta}
	nonaligning_region_file=${nonaligning_region_file_temp/#${Extra
ct_region_path}\//}
	perl
~/Documents/Projects/UtilityScripts/scripts_by_Joe/translate_from_all
_ATG_v2.pl ${nonaligning_region}
${nonaligning_region_file}_translation.fasta
done

echo "Script finished"
```

extract_nonaligning_regions_QRY.sh

```
#!/bin/bash

#This bash script takes as input a fasta genome, and a coordinate
file (.coords) generated by nucmer
```

```
#Call this script by using: bash extract_nonaligning_regions_QRY.sh
<COORDS FILE> <GENOME FASTA FILE>
#This script should be run from inside the Analysis directory (the
directory containing the nucmer alignment .coord files) if run
outside a pipeline

#By Angus Malmgren, TSL

coords_in_file=$1
genome_in_file=$2
main_DIR=$3
echo "Input files to sequence extractor script:"
echo "Coordinate file: ${coords_in_file}"
echo "Genome fasta file: ${genome_in_file}"

#The name of the query genome directory:
QRY_genome_DIR=nucmer_alignments

#Name of the extracted regions directory:
Extracted_regions_DIR=Extracted_nonaligning_regions

#From the coordinate file extract the aligning region as a bed file -
first remove the header, then extract the relevant columns
tail -n +6 ${coords_in_file} | awk -v FS=" " -v OFS="\t" '{print
$19,$4,$5}' > ${coords_in_file%.coords}_QRY_coords.bed.temp

#Switch the order of the 2nd and 3rd column entries if the 2nd is
greater than the 3rd, then subtract 1 from the whole of the 2nd
column:
awk -v FS="\t" -v OFS="\t" '$2 > $3 { temp = $3; $3 = $2; $2 = temp }
1' ${coords_in_file%.coords}_QRY_coords.bed.temp >
${coords_in_file%.coords}_QRY_coords.bed.temp2
awk -v FS="\t" -v OFS="\t" '{print $1,($2 - 1),$3}'
${coords_in_file%.coords}_QRY_coords.bed.temp2 >
${coords_in_file%.coords}_QRY_coords.bed
rm ${coords_in_file%.coords}_QRY_coords.bed.temp
rm ${coords_in_file%.coords}_QRY_coords.bed.temp2

#Index the genome (to extract coordinates of contigs):
/Users/malmgren/Documents/Software/sam_tools/bin/samtools faidx
${genome_in_file}

#Convert the samtools faidx output to a bed-format output giving the
length of each contig:
awk 'BEGIN {FS="\t"}; {print $1 FS "0" FS $2}' ${genome_in_file}.fai
> ${genome_in_file%.fasta}_contig_coords.bed

genome_contig_coords=${genome_in_file%.fasta}_contig_coords.bed
```

```
genome_contig_coords_nopath2=${genome_contig_coords/#${main_DIR}\//}
genome_contig_coords_nopath3=${genome_contig_coords_nopath2/#${QRY_ge
nome_DIR}\//}
genome_contig_coords_nopath=${genome_contig_coords_nopath3/#QUERY_gen
omes\//}

echo "Genome contig coords (no path):"
echo ${genome_contig_coords_nopath}
echo "Directory containing extracted regions:"
echo ${Extracted_regions_DIR}

##Extract aligning regions:
bedtools getfasta -fi ${genome_in_file} -bed
${coords_in_file%.coords}_QRY_coords.bed -fo
${main_DIR}/${Extracted_regions_DIR}/${genome_contig_coords_nopath%_c
ontig_coords.bed}_aligning_region.fasta

##Extract non-aligning regions (query genome doesn't align to the
reference genome at given filtering threshold):
#First, use bedtools subtract to get the coordinates of the non-
aligning region of the genome:
     #a = reference, b = aligning region bed
bedtools subtract -a ${genome_in_file%.fasta}_contig_coords.bed -b
${coords_in_file%.coords}_QRY_coords.bed >
${main_DIR}/${Extracted_regions_DIR}/${genome_contig_coords_nopath%_c
ontig_coords.bed}_non_aligning_region.bed

#Extract the region based on the coordinates:
bedtools getfasta -fi ${genome_in_file} -bed
${main_DIR}/${Extracted_regions_DIR}/${genome_contig_coords_nopath%_c
ontig_coords.bed}_non_aligning_region.bed -fo
${main_DIR}/${Extracted_regions_DIR}/${genome_contig_coords_nopath%_c
ontig_coords.bed}_non_aligning_region.fasta
```

Mini-chromosome mapping

This method was used for displaying mini-chromosome sequencing read depth along contigs, with much advice and code generously provided by members of the Kamoun lab, in particular Thorsten Langner.

The general process is as follows:

1. Mini-chromosome read mapping (bwa mem, samtools (view, sort, index, idxstats) + bwa index of genome)
2. samtools view (use -F option) for filtering out non-unique/low quality reads
3. Bedtools (makewindows, coverage, groupby (mean))
4. Circlize plotting

Below, I discuss the method used, with an example given for I performed this analysis with Italian rice-infecting isolates, although the same method and commands were used for the work described in this thesis.

Initially, perform indexing with the `bwa index` command. Then, run the main processing with: `bwa mem` using the parameters `-t 4 -k 15`; followed by conversion to a .bam file using `samtools view -b -o`. This should be followed by using `samtools sort` to sort by the left-most coordinate. Next, `samtools index` should be used to index the sorted file, and statistics reported with `samtools idxstats`.

The .idxstats output file should have the following format:

```
PR003_Contig01  6063740 25407521        0
PR003_Contig02  8582813 19244737        0
```

To compare the number of mapped and unmapped reads, use the following commands on the command line:
To obtain the number of mapped reads, use:

```
awk '{sum+=$3;} END{print sum;}'
All_reads_trimmed_mini014_pr003_bwa_mem_ref_sorted.bam.idxstats
```

To obtain the number of unmapped reads, use:

```
awk '{sum+=$4;} END{print sum;}'
All_reads_trimmed_mini014_pr003_bwa_mem_ref_sorted.bam.idxstats
```

If filtering to extract uniquely mapping reads with samtools, do it at this point with the following code (kindly provided by Thorsten Langner):

```
samtools view -h -q 1 -F 4 -F 256 ${1} | grep -v XA:Z | grep -v SA:Z
| samtools view -b - > `basename ${1}
.bam`_unique_q1_primary_mapped.bam
```

Following mapping, windows need to be created for visualisation.
This stage needs a different genome indexing software to be used, generating a .fai, using: `samtools faidx`.

Then run the following to generate windows (modified based on code provided by Thorsten Langner):

```
bedtools makewindows -g ${1}.fai -w 1000 -s 500 | bedtools coverage -
d -abam ${2} -b stdin | bedtools groupby -g 1,2,3 -c 5 -o mean >
`basename ${2} .algn.bam`_vs_Ref_cov_sliding_Wind.txt
```

For the .fai files, use the following awk command to get the 1st and 2nd columns, and add a '1' in between (to get it into a format that the Circlize R library can use for plotting):

```
for i in *.fai; do awk '{print $1,"1",$2}' $i >${i}.plotting.txt ;
done
```

I used the following to sort the window files first by column 1, then by column 2:

```
sort -k1.13,1n -k2,2n -o
All_reads_trimmed_mini014_pr003_bwa_mem_ref_sorted.bam_vs_Ref_cov_sli
```

```
ding_Wind.txt.sorted.txt
All_reads_trimmed_mini014_pr003_bwa_mem_ref_sorted.bam_vs_Ref_cov_sli
ding_Wind.txt
```
Please note that in the above command, the `13`, in `-k1.13` may need changing depending on how many characters are in the contig name. Otherwise, the reordering will not function correctly.

The output should have the format:
```
PR003_Contig01  0       1000    929.546000000000049113
PR003_Contig01  500     1500    768.91899999999998272
```

I used the following code to create a file for contigs of size <= 2Mb:
```
for i in *.fai.plotting.txt; do awk '$3<=2000000' $i
>${i}.plotting.lessThanOrEqTo2mb.txt ; done
```

I also did the same for size >2Mb:
```
for i in *.fai.plotting.txt; do awk '$3>2000000' $i
>${i}.plotting.greaterThan2mb.txt ; done
```

I also did this for size <=200kb:
```
for i in *.fai.plotting.txt; do awk '$3<=200000' $i
>${i}.plotting.lessThanOrEqTo200kb.txt ; done
```

This can be then used in the R Circlize plotting script described in the section entitled ‘figure generation’.

RepeatMasker

This method was used for showing repeat content on tracks alongside mini-chromosome read depth in Circos-style plots using the Circlize R library. Advice and guidance were provided by Thorsten Langner and Joe Win.

This should be applied to the genome fasta files, to create a masked version of the genomes where repeats are masked out by ‘N’ (subsequent steps put the data into a format where it can be plotted using Circlize against the genome):

Ensure the repeat library (in this case, nonRed_repeats.B71.MOD.2.fa was used) is in the same directory as the script and data files.

Using the command `RepeatMasker -lib RepeatLibrary.fa genome.fasta` allows running RepeatMasker using a custom repeat library. The outputs will be a masked file, a map file, and an overview table file.

Next, use awk to extract (from the `fast.out` file) only columns 5,6, and 7 (to put into a bed-like format):
```
for file in *.fasta.out; do awk '{print $5"\t"$6"\t" $7}' ${file} >
${file}.bed; done
```

If indexing the genome has not yet been done, run the command `samtools faidx`, giving the fasta file as an argument.

The `.bam` files should be edited to remove the non-data header rows – typically the first three rows. For larger numbers of files, use the command `for file in *.out.bed; do sed -i 1,3d ${file}; done`.

To generate windows, the following command should be used, taking as input the indexed genome file output by `faidx` (`.fai`), and the `.bam` file after using samtools for filtering for uniquely mapping reads:

```
bedtools makewindows -g ${1}.fai -w 1000 -s 500 | bedtools coverage -d -a ${2} -b stdin | bedtools groupby -g 1,2,3 -c 5 -o sum > `basename ${2}`_vs_Ref_cov_sliding_Wind_Repeat_content.txt
```

It is then important to sort the outputs (as discussed in the previous section, the `-k1.13` will likely need to be modified for different datasets corresponding to the length of contig names - in this case the `13` refers to the 14 characters of the contig name):

```
for file in *.txt; do sort -k1.13,1n -k2,2n  -o ${file}.sorted.txt ${file}; done
```

Finally, to generate a coordinate file for plotting with Circlize, use: `for i in *.fai; do awk '{print $1,"1",$2}' $i >${i}.plotting.txt ; done`

Multiple files corresponding to contigs of different size groupings can be generated with the following:

For contigs of size <= 2Mb: `for i in *.fai.plotting.txt; do awk '$3<=2000000' $i >${i}.plotting.lessThanOrEqTo2mb.txt ; done`

For size >2Mb: `for i in *.fai.plotting.txt; do awk '$3>2000000' $i >${i}.plotting.greaterThan2mb.txt ; done`

Then, run the R script as shown in the following section:

`Circlize_plotting_script_Italian_genome_mini_split.R`

Figure generation

Two scripts are provided in this section:

- `Circlize_plotting_script_Italian_genome_mini_split.R`
  - For generating Circlize plots of mini-chromosome coverage for Italian isolates
- `horizontal_mini_coverage_plotting_v2.py`
  - Used to generate a horizontal plot of mini-chromosome coverage for AG006 (Italian) mini-chromosome contigs.

Circlize_plotting_script_Italian_genome_mini_split.R

```
library(circlize)

setwd("/path/to/working/directory")
```

```
#########################################################
##############                                Read in data:
# Read in the edited .fai index files:

Genome_AG006_small <-
read.delim("../FG1846_04_pilon_polished_round2_nonmito_reordered_rena
med.fasta.fai.plotting.txt.plotting.lessThanOrEqTo2mb.txt", sep = "
", header=FALSE,colClasses=c("character", "numeric", "numeric"))
Genome_AG006_large <-
read.delim("../FG1846_04_pilon_polished_round2_nonmito_reordered_rena
med.fasta.fai.plotting.txt.plotting.greaterThan2mb.txt", sep = " ",
header=FALSE,colClasses=c("character", "numeric", "numeric"))
Genome_AG006_small$V1 <-
as.factor(gsub("AG006_Contig","",Genome_AG006_small$V1))
Genome_AG006_large$V1 <-
as.factor(gsub("AG006_Contig","",Genome_AG006_large$V1))

##################
#Read in the sorted Mini-chromosome data:

#AG006:
Mini009_filtered <-
read.delim("../All_reads_trimmed_mini009_ag006_bwa_mem_ref_sorted_uni
que_q1_primary_mapped.bam_vs_Ref_cov_sliding_Wind.txt.sorted.txt",
header=FALSE)
Mini009_filtered$V1 <-
as.factor(gsub("AG006_Contig","",Mini009_filtered$V1))
Mini010_filtered <-
read.delim("../All_reads_trimmed_mini010_ag006_bwa_mem_ref_sorted_uni
que_q1_primary_mapped.bam_vs_Ref_cov_sliding_Wind.txt.sorted.txt",
header=FALSE)
Mini010_filtered$V1 <-
as.factor(gsub("AG006_Contig","",Mini010_filtered$V1))
Mini011_filtered <-
read.delim("../All_reads_trimmed_mini011_ag006_bwa_mem_ref_sorted_uni
que_q1_primary_mapped.bam_vs_Ref_cov_sliding_Wind.txt.sorted.txt",
header=FALSE)
Mini011_filtered$V1 <-
as.factor(gsub("AG006_Contig","",Mini011_filtered$V1))
Mini012_filtered <-
read.delim("../All_reads_trimmed_mini012_ag006_bwa_mem_ref_sorted_uni
que_q1_primary_mapped.bam_vs_Ref_cov_sliding_Wind.txt.sorted.txt",
header=FALSE)
Mini012_filtered$V1 <-
as.factor(gsub("AG006_Contig","",Mini012_filtered$V1))

# Read in Repeat data:
```

```
AG006_Repeats <-
read.delim("../updated_Repeat_mapping/FG1846_04_pilon_polished_round2
_nonmito_reordered_renamed.fasta.out.bed_vs_Ref_cov_sliding_Wind_Repe
at_content.txt.sorted.txt", header=FALSE)
AG006_Repeats$V1 <-
as.factor(gsub("AG006_Contig","",AG006_Repeats$V1))

#########################################################

# Optional read in of filtered Nanopore data

#BTJP_Nanopore <-
read.delim("ERR2612748_minimap2L_aln_BTJP_pilcon_polished_nonmitochon
drial_reordered_renamed_sorted_unique_q20.bam_vs_Ref_cov_sliding_Wind
_Nanopore.txt.sorted.txt", header=FALSE)
#BTJP_Nanopore$V1 <- as.factor(gsub("Contig","",BTJP_Nanopore$V1))


# Limit y axis of nanopore reads to 2x mean genomic values:
#BTJP_y_lim_nanopore=2*mean(BTJP_Nanopore$V4)
#BTJP_Nanopore_capped<-BTJP_Nanopore
#BTJP_Nanopore_capped$V4<- replace(BTJP_Nanopore_capped$V4,
BTJP_Nanopore_capped$V4>BTJP_y_lim_nanopore, BTJP_y_lim_nanopore)


###############################################
# PLOTTING                                    #
###############################################

#Plot AG006 contigs <= 2Mb, filtered:
circos.clear()
circos.par("start.degree" = 90)
circos.initializeWithIdeogram(Genome_AG006_small)
number_contigs<- nrow(Genome_AG006_small)
circos.track(ylim = c(0, 1), bg.col = c(rainbow(number_contigs, alpha
= 1)), bg.border = NA, track.height = 0.05)
circos.genomicTrackPlotRegion(Mini009_filtered, track.height = 0.10,
panel.fun = function(region, value, ...) {circos.genomicLines(region,
value, area = TRUE, col = "#EA3323")})#red")})
circos.genomicTrackPlotRegion(Mini010_filtered, track.height = 0.10,
panel.fun = function(region, value, ...) {circos.genomicLines(region,
value, area = TRUE, col = "#EA3323")})#red")})
circos.genomicTrackPlotRegion(Mini011_filtered, track.height = 0.10,
panel.fun = function(region, value, ...) {circos.genomicLines(region,
value, area = TRUE, col = "#EA3323")})#red")})
```

```
circos.genomicTrackPlotRegion(Mini012_filtered, track.height = 0.10,
panel.fun = function(region, value, ...) {circos.genomicLines(region,
value, area = TRUE, col = "#EA3323")})#red")})
# Nanopore read plotting:
#circos.genomicTrackPlotRegion(BTJP_Nanopore_capped,
ylim=c(0,BTJP_y_lim_nanopore*0.92),track.height = 0.15,panel.fun =
function(region, value, ...) {circos.genomicLines(region, value, area
= TRUE, col = "green")})
circos.genomicTrackPlotRegion(AG006_Repeats, track.height = 0.10,
panel.fun = function(region, value, ...) {circos.genomicLines(region,
value, area = TRUE, col = "blue")})#red")})
```

horizontal_mini_coverage_plotting_v2.py

```
#!/usr/bin/env python3

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns

print("Beginning plotting script:")

########### Specify contigs to use:
AG006_contigs=['AG006_Contig03','AG006_Contig04','AG006_Contig10','AG
006_Contig11']
###########

#Read in AG006 data:
AG006_coords_df=pd.read_csv("/Users/malmgren/Documents/Projects/Itali
an_isolate_project_part_2/mini_mapping/FG1846_04_pilon_polished_round
2_nonmito_reordered_renamed.fasta.fai.plotting.txt.plotting.lessThanO
rEqTo2mb.txt", delimiter=' ', names=['contig','start','end'])
AG006_coords_df=AG006_coords_df[AG006_coords_df.contig.isin(AG006_con
tigs)]

AG006_coverage_mini009_df=pd.read_csv("/Users/malmgren/Documents/Proj
ects/Italian_isolate_project_part_2/mini_mapping/All_reads_trimmed_mi
ni009_AG006_bwa_mem_ref_sorted_unique_q1_primary_mapped.bam_vs_Ref_co
v_sliding_Wind.txt.sorted.txt", delimiter='\t',
names=['contig','window_start','window_end','coverage'])
AG006_coverage_mini009_df=AG006_coverage_mini009_df[AG006_coverage_mi
ni009_df.contig.isin(AG006_contigs)]
AG006_coverage_mini010_df=pd.read_csv("/Users/malmgren/Documents/Proj
ects/Italian_isolate_project_part_2/mini_mapping/All_reads_trimmed_mi
ni010_AG006_bwa_mem_ref_sorted_unique_q1_primary_mapped.bam_vs_Ref_co
v_sliding_Wind.txt.sorted.txt", delimiter='\t',
names=['contig','window_start','window_end','coverage'])
```

```
AG006_coverage_mini010_df=AG006_coverage_mini010_df[AG006_coverage_mi
ni010_df.contig.isin(AG006_contigs)]
AG006_coverage_mini011_df=pd.read_csv("/Users/malmgren/Documents/Proj
ects/Italian_isolate_project_part_2/mini_mapping/All_reads_trimmed_mi
ni011_AG006_bwa_mem_ref_sorted_unique_q1_primary_mapped.bam_vs_Ref_co
v_sliding_Wind.txt.sorted.txt", delimiter='\t',
names=['contig','window_start','window_end','coverage'])
AG006_coverage_mini011_df=AG006_coverage_mini011_df[AG006_coverage_mi
ni011_df.contig.isin(AG006_contigs)]
AG006_coverage_mini012_df=pd.read_csv("/Users/malmgren/Documents/Proj
ects/Italian_isolate_project_part_2/mini_mapping/All_reads_trimmed_mi
ni012_AG006_bwa_mem_ref_sorted_unique_q1_primary_mapped.bam_vs_Ref_co
v_sliding_Wind.txt.sorted.txt", delimiter='\t',
names=['contig','window_start','window_end','coverage'])
AG006_coverage_mini012_df=AG006_coverage_mini012_df[AG006_coverage_mi
ni012_df.contig.isin(AG006_contigs)]
print(AG006_coverage_mini009_df)
print(AG006_coverage_mini010_df)
print(AG006_coverage_mini011_df)
print(AG006_coverage_mini012_df)
print()

#Plotting:
sns.set_style("white")

#Plot AG006:
x_vals_m09_c03=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig03"]["window_end"]
y_vals_m09_c03=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig03"]["coverage"]
x_vals_m09_c04=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig04"]["window_end"]
y_vals_m09_c04=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig04"]["coverage"]
x_vals_m09_c10=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig10"]["window_end"]
y_vals_m09_c10=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig10"]["coverage"]
x_vals_m09_c11=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig11"]["window_end"]
y_vals_m09_c11=AG006_coverage_mini009_df[AG006_coverage_mini009_df["c
ontig"]=="AG006_Contig11"]["coverage"]

x_vals_m10_c03=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig03"]["window_end"]
y_vals_m10_c03=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig03"]["coverage"]
```

```
x_vals_m10_c04=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig04"]["window_end"]
y_vals_m10_c04=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig04"]["coverage"]
x_vals_m10_c10=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig10"]["window_end"]
y_vals_m10_c10=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig10"]["coverage"]
x_vals_m10_c11=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig11"]["window_end"]
y_vals_m10_c11=AG006_coverage_mini010_df[AG006_coverage_mini010_df["c
ontig"]=="AG006_Contig11"]["coverage"]

x_vals_m11_c03=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig03"]["window_end"]
y_vals_m11_c03=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig03"]["coverage"]
x_vals_m11_c04=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig04"]["window_end"]
y_vals_m11_c04=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig04"]["coverage"]
x_vals_m11_c10=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig10"]["window_end"]
y_vals_m11_c10=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig10"]["coverage"]
x_vals_m11_c11=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig11"]["window_end"]
y_vals_m11_c11=AG006_coverage_mini011_df[AG006_coverage_mini011_df["c
ontig"]=="AG006_Contig11"]["coverage"]

x_vals_m12_c03=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig03"]["window_end"]
y_vals_m12_c03=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig03"]["coverage"]
x_vals_m12_c04=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig04"]["window_end"]
y_vals_m12_c04=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig04"]["coverage"]
x_vals_m12_c10=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig10"]["window_end"]
y_vals_m12_c10=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig10"]["coverage"]
x_vals_m12_c11=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig11"]["window_end"]
y_vals_m12_c11=AG006_coverage_mini012_df[AG006_coverage_mini012_df["c
ontig"]=="AG006_Contig11"]["coverage"]

#Count number of windows in each contig, to allow scaling of axes:
```

```
AG006_contig03_len=AG006_coverage_mini009_df[AG006_coverage_mini009_d
f.contig=='AG006_Contig03'].shape[0]
AG006_contig04_len=AG006_coverage_mini009_df[AG006_coverage_mini009_d
f.contig=='AG006_Contig04'].shape[0]
AG006_contig10_len=AG006_coverage_mini009_df[AG006_coverage_mini009_d
f.contig=='AG006_Contig10'].shape[0]
AG006_contig11_len=AG006_coverage_mini009_df[AG006_coverage_mini009_d
f.contig=='AG006_Contig11'].shape[0]

AG006_mini009_max=AG006_coverage_mini009_df[AG006_coverage_mini009_df
['contig'].isin(['AG006_Contig11', 'AG006_Contig10',
'AG006_Contig04', 'AG006_Contig03'])]['coverage'].max()
print('max:', AG006_mini009_max)
AG006_mini010_max=AG006_coverage_mini010_df[AG006_coverage_mini009_df
['contig'].isin(['AG006_Contig11', 'AG006_Contig10',
'AG006_Contig04', 'AG006_Contig03'])]['coverage'].max()
print('max:', AG006_mini010_max)
AG006_mini011_max=AG006_coverage_mini011_df[AG006_coverage_mini009_df
['contig'].isin(['AG006_Contig11', 'AG006_Contig10',
'AG006_Contig04', 'AG006_Contig03'])]['coverage'].max()
print('max:', AG006_mini011_max)
AG006_mini012_max=AG006_coverage_mini012_df[AG006_coverage_mini009_df
['contig'].isin(['AG006_Contig11', 'AG006_Contig10',
'AG006_Contig04', 'AG006_Contig03'])]['coverage'].max()
print('max:', AG006_mini012_max)

#Get nearest 100:
AG006_mini009_max_nearest100=(int(AG006_mini009_max/100.0)+1)*100.0
AG006_mini010_max_nearest100=(int(AG006_mini010_max/100.0)+1)*100.0
AG006_mini011_max_nearest100=(int(AG006_mini011_max/100.0)+1)*100.0
AG006_mini012_max_nearest100=(int(AG006_mini012_max/100.0)+1)*100.0

#Use the following to get common y-axis: fig,axs = plt.subplots(4,4,
sharex='col', sharey=True,
fig,axs = plt.subplots(4,4, sharex='col', sharey='row',
gridspec_kw={'hspace':0.1, 'wspace':0.05,
'width_ratios':[AG006_contig03_len,AG006_contig04_len,AG006_contig10_
len,AG006_contig11_len]}, figsize=(40,8))
fig.suptitle('AG006', fontsize=30)
axs[0,0].plot(x_vals_m09_c03, y_vals_m09_c03,color='black')
axs[0,0].set_ylim([0,AG006_mini009_max_nearest100])
axs[0,1].plot(x_vals_m09_c04, y_vals_m09_c04,color='black')
axs[0,2].plot(x_vals_m09_c10, y_vals_m09_c10,color='black')
axs[0,3].plot(x_vals_m09_c11, y_vals_m09_c11,color='black')

axs[0,0].fill_between(x_vals_m09_c03, 0, y_vals_m09_c03, color='red')
axs[0,1].fill_between(x_vals_m09_c04, 0, y_vals_m09_c04, color='red')
axs[0,2].fill_between(x_vals_m09_c10, 0, y_vals_m09_c10, color='red')
```

```
axs[0,3].fill_between(x_vals_m09_c11, 0, y_vals_m09_c11, color='red')

axs[0,0].set_title('Contig03', fontsize=20)
axs[0,1].set_title('Contig04', fontsize=20)
axs[0,2].set_title('Contig10', fontsize=20)
axs[0,3].set_title('Contig11', fontsize=20)

axs[1,0].plot(x_vals_m10_c03, y_vals_m10_c03,color='black')
axs[1,0].set_ylim([0,AG006_mini010_max_nearest100])
axs[1,1].plot(x_vals_m10_c04, y_vals_m10_c04,color='black')
axs[1,2].plot(x_vals_m10_c10, y_vals_m10_c10,color='black')
axs[1,3].plot(x_vals_m10_c11, y_vals_m10_c11,color='black')

axs[1,0].fill_between(x_vals_m10_c03, 0, y_vals_m10_c03, color='red')
axs[1,1].fill_between(x_vals_m10_c04, 0, y_vals_m10_c04, color='red')
axs[1,2].fill_between(x_vals_m10_c10, 0, y_vals_m10_c10, color='red')
axs[1,3].fill_between(x_vals_m10_c11, 0, y_vals_m10_c11, color='red')

axs[2,0].plot(x_vals_m11_c03, y_vals_m11_c03,color='black')
axs[2,0].set_ylim([0,AG006_mini011_max_nearest100])
axs[2,1].plot(x_vals_m11_c04, y_vals_m11_c04,color='black')
axs[2,2].plot(x_vals_m11_c10, y_vals_m11_c10,color='black')
axs[2,3].plot(x_vals_m11_c11, y_vals_m11_c11,color='black')

axs[2,0].fill_between(x_vals_m11_c03, 0, y_vals_m11_c03, color='red')
axs[2,1].fill_between(x_vals_m11_c04, 0, y_vals_m11_c04, color='red')
axs[2,2].fill_between(x_vals_m11_c10, 0, y_vals_m11_c10, color='red')
axs[2,3].fill_between(x_vals_m11_c11, 0, y_vals_m11_c11, color='red')

axs[3,0].plot(x_vals_m12_c03, y_vals_m12_c03,color='black')
axs[3,0].set_ylim([0,AG006_mini012_max_nearest100])
axs[3,1].plot(x_vals_m12_c04, y_vals_m12_c04,color='black')
axs[3,2].plot(x_vals_m12_c10, y_vals_m12_c10,color='black')
axs[3,3].plot(x_vals_m12_c11, y_vals_m12_c11,color='black')

axs[3,0].xaxis.major.formatter._useMathText = True
axs[3,1].xaxis.major.formatter._useMathText = True
axs[3,2].xaxis.major.formatter._useMathText = True
axs[3,3].xaxis.major.formatter._useMathText = True

axs[3,0].fill_between(x_vals_m12_c03, 0, y_vals_m12_c03, color='red')
axs[3,1].fill_between(x_vals_m12_c04, 0, y_vals_m12_c04, color='red')
axs[3,2].fill_between(x_vals_m12_c10, 0, y_vals_m12_c10, color='red')
axs[3,3].fill_between(x_vals_m12_c11, 0, y_vals_m12_c11, color='red')

#To remove axis labels from 'inner' plots:
for ax in axs.flat:
     ax.label_outer()
```

```
plt.savefig('Horizontal_mini_coverage_AG006.pdf')
```

Appendix 2 – Sequences

Contigs designated as mini-chromosome contigs in the three Bangladeshi assemblies:

BTMP-S13-1	**BGP1-b**	**BTJP4-1**
Contig10	Contig14	Contig13
Contig11	Contig20	Contig24
Contig12	Contig22	Contig25
	Contig34	Contig26
	Contig35	Contig28
	Contig36	Contig36
	Contig37	Contig37
	Contig38	Contig39
	Contig39	Contig41
	Contig42	Contig49
	Contig47	Contig52
	Contig51	
	Contig63	
	Contig66	
	Contig67	
	Contig72	

The Art1_WB_ZM and APiasL3 sequences are given below, with signal peptide sequences marked in grey.

>Art1_WB_ZM
MHGSIVKLALFLSLSNIASAVPNMVYFYTDRNADAVDAQDGMHKGDLMWPELSPLTRAATA
AAGKYWIYHIRTSGIDERFVEKDGKWEATGKIPYESIAGWDFFYKSRGRVLTIFHQNDGRRA
RDTFTRRIE

>APiasL3
MRFSPIPILFLTTGVIAAQIQDSSTRNPDHIVKCNTKLSSRKLERRNEDYWACLRICVAGGIYG
ALKFSNITVRDTANCATACAAVFGCPNI

APiasL family

Sequences for representative APiasL family members are given with signal peptide sequences removed.
>AVR-Pias
QIQDSSTKNPNQVVNRNAKLSSRNLERRNENYWRCVNICIAGGVFGALKFTDITVRDSVHCA
GACAAVFGYPE
>APiasL2
TPIRDSSIENPNHVVNRNTKLSGRDLERRNDAYVKCIGACLAGLGVFGAANFIELKLRNYADC
ATVCGLVFLN
>APiasL3
AQIQDSSTRNPDHIVKCNTKLSSRKLERRNEDYWACLRICVAGGIYGALKFSNITVRDTANCA
TACAAVFGCPNI
>APiasL4

QIQDSSTKNPNQIVKCNTKLSTRNLERRNDDYWQCLRICVAGGVFGALRFTNLTIRDAANCA
TACAAVFGCP
>APiasL5
AQIQDSSIQNPNHVAKCKTKLSNRSLERRNENYYECVRICIAGGAFGALTFGNISIRDVKNCA
GVCEKFFGPPDA
>APiasL6
AQIQDSSSQNPNHVAKCNKKLSNRSLERRNENYYACLRICIAGGVYGALRFTDITVRDAVNC
ATACAAVFGAPNI
>APiasL7
QIQDFFIKNPNYIIKCNTKFSNRKFKRRNEDYWVYLKICVAGGI
>APiasL8
FQASNLEHDAKYDSKLSTRGLERRDSDYATCVKVCIIGGVLGYFNFVNIGPRDVVTCEGVCD
KIFGLR
>APiasL9
SPVHYLQTTDFGHITKRNSELLTRGLQRRDEGFYPCVAACVAGGVYMNLKFTDHGPRPRNF
CINTCADVYGLPENLRF

Plasmids for agroinfiltration:

ID	Designation	comments	Details
pAM1	Art1_WB_ZM in pUC-GW-Kan	level 0 storage: contains codon optimised effector	Storage
pAM2	APiasL3_WB_BD in pUC-GW-Kan	level 0 storage: contains codon optimised effector	Storage
pAM3	AVR-Pias in pUC-GW-Kan	level 0 storage: contains codon optimised effector	
pAM4	30005	Level 0 N terminal tag: overhangs CCAT - AATG (3x flag tag - detect with an anti-FLAG antibody)	Level 0
pAM5	41414	Level 0 Terminator: overhangs GCTT - CGCT	Level 0
pAM6	13004	Level 0 Mas promoter: overhangs GGAG - CCAT	Level 0
pAM7	47732	Level 1 acceptor:	Level 1

		overhangs GGAG - CGCT	
pAM8		The Golden Gate product from pAM4-5 and also pAM1	Agrobacterium infiltration

Primers designed for this work:

Code	Name	Sequence	Comments
oAM1	AM_Art1_WB_ZM_ZMW18_10_contig16_UF_p1	GCGTACATGCGCCTAAGC	Primer p1 (Upper Flank), for use with the split hygromycin effector deletion method, to delete Art1_WB_ZM from the genome of ZMW18_10, and replace it with Hygromycin.
oAM2	AM_Art1_WB_ZM_ZMW18_10_contig16_UF_p2	GTCGTGACTGGGAAAACCCTGGCGAGTTTAACAATGGACCCGTGC	Primer p2 for ZMW18_10, corresponding to p1. The reverse complement, with Hygromycin cassette overhang sequence added at the beginning.
oAM3	AM_Art1_WB_ZM_ZMW18_10_contig16_DF_p3	TCCTGTGTGAAATTGTTATCCGCTACTATGAGAATTAGGCCACCAGG	Primer p3 for ZMW18_10, corresponding to p4. The Hygromycin cassette overhang has been added at the beginning of the sequence.
oAM4	AM_Art1_WB_ZM_ZMW18_10_contig16_DF_p4	TACTGCTGGATACCTTTCACAGG	Primer p4 for ZMW18_10, corresponding to p3. The reverse complement of the sequence in the genome is shown here.
oAM5	AM_Art1_WB_ZM_ZMW18_10_contig16_UF_d1	GGCTCTAGTGCTGACTACGC	Diagnostic primer d1 (Upper Flank), for use with the split hygromycin deletion method, where the diagnostic primers will be used to test if integration of Hygromycin into the correct location of ZMW18_10 occurred.
oAM6	AM_Art1_WB_ZM_ZMW18_10_contig16_UF_d2	cactagctccagccaagcc	Diagnostic primer d2 (reverse complemented), selected from Hygromycin flanking region on pre-existing plasmid, used in ZMW18_10.
oAM7	AM_Art1_WB_ZM_ZMW18_10_contig16_DF_d3	ctgcaggtcgaccatatgg	Diagnostic primer d3 (forward sequence), selected from Hygromycin flanking region on pre-existing plasmid, used in ZMW18_10.
oAM8	AM_Art1_WB_ZM_ZMW18_10_contig16_DF_d4	CCCTGATGAGACATGGTATGGG	Diagnostic primer d4 (reverse complemented), selected from Hygromycin flanking region on pre-existing plasmid, used in ZMW18_10.
oAM9	AM_oAM9_seq_p1_for_pAM8	CTGGGGTGGATGCAGTGG	Sequencing primer 1 - Forward primer before promoter - These primers are designed for the Golden Gate product plasmid "pAM8", which contains Art1_WB_ZM.
oAM10	AM_oAM10_seq_p2_for_pAM8	GTGCAGAAGACAATTGCAGCG	Sequencing primer 2 - Reverse primer after terminator - These primers are designed for the Golden Gate product

			plasmid "pAM8", which contains Art1_WB_ZM.
oAM11	AM_oAM11_seq_p3_for_pAM8	GCGGTGACGCCATTTCGC	Sequencing primer 3 - Forward primer within promoter - These primers are designed for the Golden Gate product plasmid "pAM8", which contains Art1_WB_ZM.
oAM12	AM_oAM12_p1_APiasL3_UF_Contig13	CATAATAGTTTACGCGCTGGC	Primer p1 (Upper Flank), for use with the split hygromycin deletion method, to delete APiasL3 from the genome of BTJP4-1, contig13, and replace it with Hygromycin.
oAM13	AM_oAM13_p2_APiasL3_UF_Contig13	GTCGTGACTGGGAAAACCCTGGCGCTGCCGTTTTTGGTTGTCC	Primer p2, corresponding to p1, for use with BTJP4-1. The reverse complement, with Hygromycin cassette overhang sequence added at the start (as it is reverse complemented).
oAM14	AM_oAM14_p3_APiasL3_DF_Contig13	TCCTGTGTGAAATTGTTATCCGCTGATGGTTTGTTGGAGCCACG	Primer p3, corresponding to p4, for use with BTJP4-1. The Hygromycin cassette overhang has been added at the beginning of the sequence.
oAM15	AM_oAM15_p4_APiasL3_DF_Contig13	TTGTGCTTTAGGTAGTACGGG	Primer p4, corresponding to p3, for use with BTJP4-1. The reverse complement of the sequence in the genome is shown here.
oAM16	AM_oAM16_d1_v2_Art1_ZMW18_10	CAGGTTACACACGCGTTGC	Updated diagnostic primer d1 (Upper Flank), for use with the split hygromycin deletion method, where the diagnostic primers will be used to test if integration of Hygromycin into the correct location of ZMW18_10 occurred.
oAM17	AM_oAM17_d2_v2_Art1_ZMW18_10	ACAAATTTTGTGCTCACCGCC	Updated diagnostic primer d2 (reverse complemented), selected from Hygromycin flanking region on pre-existing plasmid. Designed to use with ZMW18_10.
oAM18	AM_oAM18_d3_v2_Art1_ZMW18_10	ACGAATCACTAGTGCGGCC	Updated diagnostic primer d3 (forward sequence), selected from Hygromycin flanking region on pre-existing plasmid. Designed to use with ZMW18_10.
oAM19	AM_oAM19_d4_v2_Art1_ZMW18_10	GTATCGTGGGTCGGACTCG	Updated diagnostic primer d4 (reverse complemented), designed to use with ZMW18_10.
oAM20	AM_oAM20_d1_APiasL3_UF_Contig13	TTGGCCACCGGAATAACGG	Diagnostic primer d1 (Upper Flank), for use with the split hygromycin deletion method, where the diagnostic primers will be used to test if integration of Hygromycin into the correct location of BTJP4-1 occurred. Should be used with d2 from ZMW18_10 due to shared Hygromycin cassette sequence.
oAM21	AM_oAM21_d4_APiasL3_DF_Contig13	CGAATCGTGCCTGACCGG	Diagnostic primer d4 (Upper Flank), for use with the split hygromycin deletion method (method to delete APiasL3 from

			the genome of BTJP4-1 and replace it with Hygromycin), where the diagnostic primers will be used to test if integration of Hygromycin into the correct location of BTJP4-1 occurred. Should be used with d3 from ZMW18_10 due to shared Hygromycin cassette sequence.

Appendix 3 – Additional work

Beyond the work discussed in this thesis, I have contributed to the following published studies.

"Multiple Horizontal Mini-chromosome Transfers Drive Genome Evolution of Clonal Blast Fungus Lineages"
Ana Cristina Barragan, Sergio M. Latorre, Angus Malmgren, Adeline Harant, Joe Win, Yu Sugihara, Hernán A. Burbano, Sophien Kamoun, Thorsten Langner
(Barragan *et al.*, 2024)

This paper described the discovery of horizontal transfer of a mini-chromosome into a clonal rice-infecting blast population on multiple occasions during the last 300 years, likely from *Eleusine*-infecting isolates. My contribution to this paper included preprocessing of Italian rice-infecting blast fungus genomes and identification of mini-chromosomes through mapping previously-sequenced CHEF gel data to their assemblies. I performed whole-genome alignments with Nucmer to other Italian isolates and I identified that Contig10 in the AG006 isolate did not align to sequences in the other Italian isolates. I also performed preliminary alignments using Nucmer to selected isolates in other blast fungus lineages, such as LpKY97 and wheat-infecting B71-lineage isolates. Some of these alignments indicated increased sequence alignment to AG006 Contig10 than between AG006 Contig10 and other Italian rice-infecting isolates. More structured approaches to comparing these isolates were conducted by others later on in this project, and as such, multiple aspects of the work performed by myself formed preliminary work on the project. I also applied several of the approaches described in this thesis, such as the non-aligning region identification pipeline and subsequent effector candidate identification, although the outputs from these were not used in the paper.

"Genomic surveillance uncovers a pandemic clonal lineage of the wheat blast fungus"
Sergio M. Latorre, Vincent M. Were, Andrew J.Foster, Thorsten Langner, Angus Malmgren, Adeline Harant, Soichiro Asuke, Sarai Reyes-Avila, Dipali Rani Gupta, Cassandra Jensen, Weibin Ma, Nur Uddin Mahmud, Md. Shåbab Mehebub, Rabson M. Mulenga, Abu Naim Md. Muzahid, Sanjoy Kumar Paul, S. M. Fajle Rabby, Abdullah Al Mahbub Rahat, Lauren Ryder, Ram-Krishna Shrestha, Suwilanji Sichilima, Darren M. Soanes, Pawan Kumar Singh, Alison R. Bentley, Diane G. O. Saunders, Yukio Tosa, Daniel Croll, Kurt H. Lamour, Tofazzal Islam, Batiseba Tembo, Joe Win, Nicholas J.Talbot, Hernan A. Burbano, Sophien Kamoun
(Latorre *et al.*, 2023)

This study demonstrated that the wheat-infecting blast fungus outbreaks in Zambia and Bangladesh had both originated in South America as separate introductions. It also demonstrated the effectiveness of strobilurin fungicides and the Rmg8 R gene in controlling this lineage, whilst highlighting risks of it developing fungicide resistance and recombination with endemic blast fungus lineages. I carried out data processing towards this work.

"Genomic rearrangements generate hypervariable mini-chromosomes in host-specific isolates of the blast fungus"

Thorsten Langner, Adeline Harant, Luis B. Gomez-Luciano, Ram K. Shrestha, Angus Malmgren, Sergio M. Latorre, Hernan A. Burbano, Joe Win, Sophien Kamoun
(Langner *et al.*, 2021)
In this work, my contributions were focussed particularly on figure generation and gene presence analysis. This work described structural variation identified in 4 blast fungus isolates infecting on foxtail millet, rice and goosegrass, in addition to mini-chromosomes identified within this set of isolates. Multiple genes linked to virulence were identified in a mini-chromosome found in the rice isolate, including AVR-Pik.

“Differential loss of effector genes in three recently expanded pandemic clonal lineages of the rice blast fungus”
Sergio M. Latorre, C. Sarai Reyes-Avila, Angus Malmgren, Joe Win, Sophien Kamoun and Hernán A. Burbano
(Latorre *et al.*, 2020)
In this work, I applied the machine learning technique extremely randomized trees as part of my data analysis. This study determined that the rice-infecting blast fungus lineage can be understood to form three clonal groups, likely originating within the past two centuries, alongside a recombining group centred on Southeast Asia. Further, distinct patterns of presence and absence of likely effectors can be used to distinguish between the four lineages, with the clonal lineages featuring reduced effector counts.

I also contributed to the following preprints:

“Wild grass isolates of *Magnaporthe* (Syn. *Pyricularia*) spp. from Germany can cause blast disease on cereal crops”
A. Cristina Barragan, Sergio M. Latorre, Paul G. Mock, Adeline Harant, Joe Win, Angus Malmgren, Hernán A. Burbano, Sophien Kamoun and Thorsten Langner
(Barragan *et al.*, 2022)

“SNP calling parameters have minimal impact on population structure and divergence time estimates for the rice blast fungus”
Sergio M. Latorre, Thorsten Langner, Angus Malmgren, Joe Win, Sophien Kamoun, Hernán A. Burbano
(Latorre *et al.*, 2022)

“A pandemic clonal lineage of the wheat blast fungus”
Joe Win, Angus Malmgren, Thorsten Langner, Sophien Kamoun
(Win *et al.*, 2021)

“Large scale genome assemblies of *Magnaporthe oryzae* rice isolates from Italy”
Joe Win, Adeline Harant, Angus Malmgren, Thorsten Langner, Ram-Krishna Shrestha, Sergio M. Latorre, Vincent Were, Nicholas J. Talbot, Hernán A. Burbano, Anna Maria Picco, Sophien Kamoun
(Win *et al.*, 2020)

In addition to the above, I helped preparation of rice-infecting blast fungus isolate assemblies from Italy. This work reported sequencing and polishing of nine such isolates followed by removal of mitochondrial sequences, to obtain chromosome-level assemblies. These isolates were then used in the (Barragan *et al.*, 2024) study mentioned above.

References

Aravind, L. *et al.* (2015) ‘The Natural History of ADP-Ribosyltransferases and the ADP-Ribosylation System’, *Current topics in microbiology and immunology*, 384, pp. 3–32. Available at: https://doi.org/10.1007/82_2014_414.

Barragan, A.C. *et al.* (2022) ‘Wild grass isolates of Magnaporthe (Syn. Pyricularia) spp. from Germany can cause blast disease on cereal crops’. bioRxiv, p. 2022.08.29.505667. Available at: https://doi.org/10.1101/2022.08.29.505667.

Barragan, A.C. *et al.* (2024) ‘Multiple Horizontal Mini-chromosome Transfers Drive Genome Evolution of Clonal Blast Fungus Lineages’, *Molecular Biology and Evolution*, 41(8), p. msae164. Available at: https://doi.org/10.1093/molbev/msae164.

Bendahmane, A. *et al.* (1995) ‘The coat protein of potato virus X is a strain-specific elicitor of Rx1-mediated virus resistance in potato’, *The Plant Journal*, 8(6), pp. 933–941. Available at: https://doi.org/10.1046/j.1365-313X.1995.8060933.x.

Bertazzoni, S. *et al.* (2018) ‘Accessories Make the Outfit: Accessory Chromosomes and Other Dispensable DNA Regions in Plant-Pathogenic Fungi’, *Molecular Plant-Microbe Interactions®*, 31(8), pp. 779–788. Available at: https://doi.org/10.1094/MPMI-06-17-0135-FI.

Biswas, C.S. *et al.* (2020) ‘Screening and Biochemical Characterization of Wheat Cultivars Resistance to Magnaporthe oryzae pv Triticum (MoT)’, *Journal of Plant Stress Physiology*, pp. 1–6. Available at: https://doi.org/10.25081/jpsp.2020.v6.6096.

Bos, J.I.B. *et al.* (2006) ‘The C-terminal half of Phytophthora infestans RXLR effector AVR3a is sufficient to trigger R3a-mediated hypersensitivity and suppress INF1-induced cell death in Nicotiana benthamiana’, *The Plant Journal*, 48(2), pp. 165–176. Available at: https://doi.org/10.1111/j.1365-313X.2006.02866.x.

Buchfink, B., Reuter, K. and Drost, H.-G. (2021) ‘Sensitive protein alignments at tree-of-life scale using DIAMOND’, *Nature Methods*, 18(4), pp. 366–368. Available at: https://doi.org/10.1038/s41592-021-01101-x.

Camacho, C. *et al.* (2009) ‘BLAST+: architecture and applications’, *BMC Bioinformatics*, 10(1), p. 421. Available at: https://doi.org/10.1186/1471-2105-10-421.

Ceresini, P.C. *et al.* (2018) ‘Wheat blast: from its origins in South America to its emergence as a global threat’, *Molecular Plant Pathology*, 20(2), pp. 155–172. Available at: https://doi.org/10.1111/mpp.12747.

Coelho, M.A. de O. *et al.* (2016) ‘Sowing date reduces the incidence of wheat blast disease’, *Pesquisa Agropecuária Brasileira*, 51, pp. 631–637. Available at: https://doi.org/10.1590/S0100-204X2016000500025.

Coleman, J.J. *et al.* (2009) ‘The Genome of Nectria haematococca: Contribution of Supernumerary Chromosomes to Gene Expansion’, *PLOS Genetics*, 5(8), p. e1000618. Available at: https://doi.org/10.1371/journal.pgen.1000618.

Croll, D. and McDonald, B.A. (2012) 'The Accessory Genome as a Cradle for Adaptive Evolution in Pathogens', *PLoS Pathogens*. Edited by J. Heitman, 8(4), p. e1002608. Available at: https://doi.org/10.1371/journal.ppat.1002608.

Cruz, C.D. and Valent, B. (2017) 'Wheat blast disease: danger on the move', *Tropical Plant Pathology*, 42(3), pp. 210–222. Available at: https://doi.org/10.1007/s40858-017-0159-z.

Dean, R. *et al.* (2012) 'The Top 10 fungal pathogens in molecular plant pathology', *Molecular Plant Pathology*, 13(4), pp. 414–430. Available at: https://doi.org/10.1111/j.1364-3703.2011.00783.x.

Dong, S., Raffaele, S. and Kamoun, S. (2015) 'The two-speed genomes of filamentous pathogens: waltz with plants', *Current Opinion in Genetics & Development*, 35, pp. 57–65. Available at: https://doi.org/10.1016/j.gde.2015.09.001.

Fisher, M.C. *et al.* (2012) 'Emerging fungal threats to animal, plant and ecosystem health', *Nature*, 484(7393), pp. 186–194. Available at: https://doi.org/10.1038/nature10947.

Fones, H.N. *et al.* (2020) 'Threats to global food security from emerging fungal and oomycete crop pathogens', *Nature Food*, 1(6), pp. 332–342. Available at: https://doi.org/10.1038/s43016-020-0075-0.

Gabriel, W., Lynch, M. and Bürger, R. (1993) 'MULLER'S RATCHET AND MUTATIONAL MELTDOWNS', *Evolution*, 47(6), pp. 1744–1757. Available at: https://doi.org/10.1111/j.1558-5646.1993.tb01266.x.

Gardner, S.N., Slezak, T. and Hall, B.G. (2015) 'kSNP3.0: SNP detection and phylogenetic analysis of genomes without genome alignment or reference genome', *Bioinformatics*, 31(17), pp. 2877–2878. Available at: https://doi.org/10.1093/bioinformatics/btv271.

Gibrat, J.-F., Madej, T. and Bryant, S.H. (1996) 'Surprising similarities in structure comparison', *Current Opinion in Structural Biology*, 6(3), pp. 377–385. Available at: https://doi.org/10.1016/S0959-440X(96)80058-3.

Gladieux, P., Ravel, S., *et al.* (2018) 'Coexistence of Multiple Endemic and Pandemic Lineages of the Rice Blast Pathogen', *mBio*, 9(2), p. 10.1128/mbio.01806-17. Available at: https://doi.org/10.1128/mbio.01806-17.

Gladieux, P., Condon, B., *et al.* (2018) 'Gene Flow between Divergent Cereal- and Grass-Specific Lineages of the Rice Blast Fungus Magnaporthe oryzae', *mBio*, 9(1), p. 10.1128/mbio.01219-17. Available at: https://doi.org/10.1128/mbio.01219-17.

Goswami, R.S. (2012) 'Targeted Gene Replacement in Fungi Using a Split-Marker Approach', in M.D. Bolton and B.P.H.J. Thomma (eds) *Plant Fungal Pathogens: Methods and Protocols*. Totowa, NJ: Humana Press (Methods in Molecular Biology), pp. 255–269. Available at: https://doi.org/10.1007/978-1-61779-501-5_16.

Goulart, A. and Paiva, F. (1990) 'Transmissão de Pyricularia oryzae através de sementes de trigo (Triticum aestivum)', *Fitopatologia Brasileira*, 15, pp. 359–362.

Gyawali, N. *et al.* (2023) ‘Using recurrent neural networks to detect supernumerary chromosomes in fungal strains causing blast diseases’. bioRxiv, p. 2023.09.17.558148. Available at: https://doi.org/10.1101/2023.09.17.558148.

Haeussler, S. and Langner, T. (2025) ‘Transformation of Magnaporthe oryzae’. Available at: https://www.protocols.io/view/transformation-of-magnaporthe-oryzae-d7p89mrw (Accessed: 29 September 2025).

Hartl, D.L. and Cochrane, B.J. (2019) *Genetics: Analysis of Genes and Genomes*. Ninth Edition. Jones & Bartlett Learning.

Inoue, Y. *et al.* (2017) ‘Evolution of the wheat blast fungus through functional losses in a host specificity determinant’, *Science*, 357(6346), pp. 80–83. Available at: https://doi.org/10.1126/science.aam9654.

Inoue, Y. *et al.* (2021) ‘Suppression of wheat blast resistance by an effector of Pyricularia oryzae is counteracted by a host specificity resistance gene in wheat’, *New Phytologist*, 229(1), pp. 488–500. Available at: https://doi.org/10.1111/nph.16894.

Islam, M.T. *et al.* (2016) ‘Emergence of wheat blast in Bangladesh was caused by a South American lineage of Magnaporthe oryzae’, *BMC Biology*, 14(1), p. 84. Available at: https://doi.org/10.1186/s12915-016-0309-7.

Islam, M.T. *et al.* (2020) ‘Wheat blast: a new threat to food security’, *Phytopathology Research*, 2(1), p. 28. Available at: https://doi.org/10.1186/s42483-020-00067-6.

Jeong, B. *et al.* (2011) ‘Structure Function Analysis of an ADP-ribosyltransferase Type III Effector and Its RNA-binding Target in Plant Immunity’, *The Journal of Biological Chemistry*, 286(50), pp. 43272–43281. Available at: https://doi.org/10.1074/jbc.M111.290122.

Jones, J.D.G., Vance, R.E. and Dangl, J.L. (2016) ‘Intracellular innate immune surveillance devices in plants and animals’, *Science*, 354(6316), p. aaf6395. Available at: https://doi.org/10.1126/science.aaf6395.

Kamoun, S., Talbot, N.J. and Islam, M.T. (2019) ‘Plant health emergencies demand open science: Tackling a cereal killer on the run’, *PLOS Biology*, 17(6), p. e3000302. Available at: https://doi.org/10.1371/journal.pbio.3000302.

Kanzaki, H. *et al.* (2012) ‘Arms race co-evolution of Magnaporthe oryzae AVR-Pik and rice Pik genes driven by their physical interactions’, *The Plant Journal*, 72(6), pp. 894–907. Available at: https://doi.org/10.1111/j.1365-313X.2012.05110.x.

Kobayashi, N. *et al.* (2023) ‘Horizontally Transferred DNA in the Genome of the Fungus Pyricularia oryzae is Associated With Repressive Histone Modifications’, *Molecular Biology and Evolution*, 40(9), p. msad186. Available at: https://doi.org/10.1093/molbev/msad186.

Kosakovsky Pond, S.L. *et al.* (2020) ‘HyPhy 2.5—A Customizable Platform for Evolutionary Hypothesis Testing Using Phylogenies’, *Molecular Biology and Evolution*, 37(1), pp. 295–299. Available at: https://doi.org/10.1093/molbev/msz197.

Kowalczyk, A., Chikina, M. and Clark, N.L. (2021) ‘A cautionary tale on proper use of branch-site models to detect convergent positive selection’. bioRxiv, p. 2021.10.26.465984. Available at: https://doi.org/10.1101/2021.10.26.465984.

Kryazhimskiy, S. and Plotkin, J.B. (2008) ‘The Population Genetics of dN/dS’, *PLOS Genetics*, 4(12), p. e1000304. Available at: https://doi.org/10.1371/journal.pgen.1000304.

Kusaba, M. *et al.* (2014) ‘Loss of a 1.6 Mb chromosome in Pyricularia oryzae harboring two alleles of AvrPik leads to acquisition of virulence to rice cultivars containing resistance alleles at the Pik locus’, *Current Genetics*, 60(4), pp. 315–325. Available at: https://doi.org/10.1007/s00294-014-0437-y.

Langner, T. *et al.* (2021) ‘Genomic rearrangements generate hypervariable mini-chromosomes in host-specific isolates of the blast fungus’, *PLOS Genetics*, 17(2), p. e1009386. Available at: https://doi.org/10.1371/journal.pgen.1009386.

Latorre, S.M. *et al.* (2020) ‘Differential loss of effector genes in three recently expanded pandemic clonal lineages of the rice blast fungus’, *BMC Biology*, 18(1), p. 88. Available at: https://doi.org/10.1186/s12915-020-00818-z.

Latorre, S.M. *et al.* (2022) ‘SNP calling parameters have minimal impact on population structure and divergence time estimates for the rice blast fungus’. bioRxiv, p. 2022.03.06.482794. Available at: https://doi.org/10.1101/2022.03.06.482794.

Latorre, S.M. *et al.* (2023) ‘Genomic surveillance uncovers a pandemic clonal lineage of the wheat blast fungus’, *PLOS Biology*, 21(4), p. e3002052. Available at: https://doi.org/10.1371/journal.pbio.3002052.

Latorre, S.M. and Burbano, H.A. (2021) *The emergence of wheat blast in Zambia and Bangladesh was caused by the same genetic lineage of Magnaporthe oryzae*. Zenodo. Available at: https://doi.org/10.5281/zenodo.4619405.

Liu, S. *et al.* (2023) ‘Rapid mini-chromosome divergence among fungal isolates causing wheat blast outbreaks in Bangladesh and Zambia’, *New Phytologist*, n/a(n/a). Available at: https://doi.org/10.1111/nph.19402.

Lynch, M. *et al.* (1993) ‘The Mutational Meltdown in Asexual Populations’, *Journal of Heredity*, 84(5), pp. 339–344. Available at: https://doi.org/10.1093/oxfordjournals.jhered.a111354.

Madhuprakash, J. *et al.* (2024) ‘A disease resistance protein triggers oligomerization of its NLR helper into a hexameric resistosome to mediate innate immunity’, *Science Advances*, 10(45), p. eadr2594. Available at: https://doi.org/10.1126/sciadv.adr2594.

Malaker, P.K. *et al.* (2016) ‘First Report of Wheat Blast Caused by Magnaporthe oryzae Pathotype triticum in Bangladesh’, *Plant Disease*, 100(11), pp. 2330–2330. Available at: https://doi.org/10.1094/PDIS-05-16-0666-PDN.

Marçais, G. *et al.* (2018) ‘MUMmer4: A fast and versatile genome alignment system’, *PLOS Computational Biology*, 14(1), p. e1005944. Available at: https://doi.org/10.1371/journal.pcbi.1005944.

Mehta, Y.R. *et al.* (1992) ‘Integrated management of major wheat diseases in Brazil: an example for the Southern Cone region of Latin America’, *Crop Protection*, 11(6), pp. 517–524. Available at: https://doi.org/10.1016/0261-2194(92)90168-5.

Mirdita, M. *et al.* (2022) ‘ColabFold: making protein folding accessible to all’, *Nature Methods*, 19(6), pp. 679–682. Available at: https://doi.org/10.1038/s41592-022-01488-1.

Molinari, C. and Talbot, N.J. (2022) ‘A Basic Guide to the Growth and Manipulation of the Blast Fungus, Magnaporthe oryzae’, *Current Protocols*, 2(8), p. e523. Available at: https://doi.org/10.1002/cpz1.523.

Mottaleb, K.A. *et al.* (2019) ‘Alternative use of wheat land to implement a potential wheat holiday as wheat blast control: In search of feasible crops in Bangladesh’, *Land Use Policy*, 82, pp. 1–12. Available at: https://doi.org/10.1016/j.landusepol.2018.11.046.

Muller, H.J. (1964) ‘The relation of recombination to mutational advance’, *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 1(1), pp. 2–9. Available at: https://doi.org/10.1016/0027-5107(64)90047-8.

Nicaise, V. *et al.* (2013) ‘Pseudomonas HopU1 modulates plant immune receptor levels by blocking the interaction of their mRNAs with GRP7’, *The EMBO Journal*, 32(5), pp. 701–712. Available at: https://doi.org/10.1038/emboj.2013.15.

Nielsen, R. and Slatkin, M. (2013) *An Introduction to Population Genetics: Theory and Applications*. Sinauer Associates, Inc.

Orbach, M.J. (1996) ‘Electrophoretic Karyotypes of *Magnaporthe grisea* Pathogens of Diverse Grasses’, *Molecular Plant-Microbe Interactions*, 9(4), p. 261. Available at: https://doi.org/10.1094/MPMI-9-0261.

Peng, Z. *et al.* (2019) ‘Effector gene reshuffling involves dispensable mini-chromosomes in the wheat blast fungus’, *PLOS Genetics*, 15(9), p. e1008272. Available at: https://doi.org/10.1371/journal.pgen.1008272.

Petit-Houdenot, Y. *et al.* (2020) ‘A Clone Resource of Magnaporthe oryzae Effectors That Share Sequence and Structural Similarities Across Host-Specific Lineages’, *Molecular Plant-Microbe Interactions®*, 33(8), pp. 1032–1035. Available at: https://doi.org/10.1094/MPMI-03-20-0052-A.

Pond, S.L.K., Frost, S.D.W. and Muse, S.V. (2005) ‘HyPhy: hypothesis testing using phylogenies’, *Bioinformatics*, 21(5), pp. 676–679. Available at: https://doi.org/10.1093/bioinformatics/bti079.

Raffaele, S. *et al.* (2010) ‘Analyses of genome architecture and gene expression reveal novel candidate virulence factors in the secretome of Phytophthora infestans’, *BMC Genomics*, 11(1), p. 637. Available at: https://doi.org/10.1186/1471-2164-11-637.

Reece, J.B. *et al.* (2011) *Campbell Biology*. Ninth Edition. Pearson.

Robinson, J.T. *et al.* (2011) 'Integrative Genomics Viewer', *Nature biotechnology*, 29(1), pp. 24–26. Available at: https://doi.org/10.1038/nbt.1754.

Saleh, D. *et al.* (2012) 'Sex at the origin: an Asian population of the rice blast fungus Magnaporthe oryzae reproduces sexually', *Molecular Ecology*, 21(6), pp. 1330–1344. Available at: https://doi.org/10.1111/j.1365-294X.2012.05469.x.

Shimizu, M. *et al.* (2022) 'A genetically linked pair of NLR immune receptors shows contrasting patterns of evolution', *Proceedings of the National Academy of Sciences*, 119(27), p. e2116896119. Available at: https://doi.org/10.1073/pnas.2116896119.

Singh, P.K. *et al.* (2021) 'Wheat Blast: A Disease Spreading by Intercontinental Jumps and Its Management Strategies', *Frontiers in Plant Science*, 12. Available at: https://www.frontiersin.org/articles/10.3389/fpls.2021.710707 (Accessed: 25 July 2023).

Takeuchi, N., Kaneko, K. and Koonin, E.V. (2014) 'Horizontal Gene Transfer Can Rescue Prokaryotes from Muller's Ratchet: Benefit of DNA from Dead Cells and Population Subdivision', *G3 Genes|Genomes|Genetics*, 4(2), pp. 325–339. Available at: https://doi.org/10.1534/g3.113.009845.

Talbot, N.J. (2003) 'On the Trail of a Cereal Killer: Exploring the Biology of Magnaporthe grisea', *Annual Review of Microbiology*, 57(Volume 57, 2003), pp. 177–202. Available at: https://doi.org/10.1146/annurev.micro.57.030502.090957.

Tameling, W.I.L. *et al.* (2010) 'RanGAP2 Mediates Nucleocytoplasmic Partitioning of the NB-LRR Immune Receptor Rx in the Solanaceae, Thereby Dictating Rx Function', *The Plant Cell*, 22(12), pp. 4176–4194. Available at: https://doi.org/10.1105/tpc.110.077461.

Tembo, B. *et al.* (2020) 'Detection and characterization of fungus (Magnaporthe oryzae pathotype Triticum) causing wheat blast disease on rain-fed grown wheat (Triticum aestivum L.) in Zambia', *PLOS ONE*, 15(9), p. e0238724. Available at: https://doi.org/10.1371/journal.pone.0238724.

Torto, T.A. *et al.* (2003) 'EST mining and functional expression assays identify extracellular effector proteins from the plant pathogen Phytophthora', *Genome Research*, 13(7), pp. 1675–1685. Available at: https://doi.org/10.1101/gr.910003.

Urashima, A.S. *et al.* (2009) 'Effect of Magnaporthe grisea on Seed Germination, Yield and Quality of Wheat', in G.-L. Wang and B. Valent (eds) *Advances in Genetics, Genomics and Control of Rice Blast Disease*. Dordrecht: Springer Netherlands, pp. 267–277. Available at: https://doi.org/10.1007/978-1-4020-9500-9_27.

Valent, B. *et al.* (2021) 'Recovery Plan for Wheat Blast Caused by Magnaporthe oryzae Pathotype Triticum', *Plant Health Progress*, 22(2), pp. 182–212. Available at: https://doi.org/10.1094/PHP-11-20-0101-RP.

Wilson, R.A. and Talbot, N.J. (2009) ‘Under pressure: investigating the biology of plant infection by Magnaporthe oryzae’, *Nature Reviews Microbiology*, 7(3), pp. 185–195. Available at: https://doi.org/10.1038/nrmicro2032.

Win, J. *et al.* (2020) ‘Large scale genome assemblies of Magnaporthe oryzae rice isolates from Italy’. Available at: https://doi.org/10.5281/zenodo.4326823.

Win, J. *et al.* (2021) ‘A pandemic clonal lineage of the wheat blast fungus’. Available at: https://doi.org/10.5281/zenodo.4618522.

Yoshida, K. *et al.* (2016) ‘Host specialization of the blast fungus Magnaporthe oryzae is associated with dynamic gain and loss of genes linked to transposable elements’, *BMC Genomics*, 17(1), p. 370. Available at: https://doi.org/10.1186/s12864-016-2690-6.

Zeigler, R.S. *et al.* (1997) ‘Evidence of Parasexual Exchange of DNA in the Rice Blast Fungus Challenges Its Exclusive Clonality’, *Phytopathology*®, 87(3), pp. 284–294. Available at: https://doi.org/10.1094/PHYTO.1997.87.3.284.

Zhang, J. (2004) ‘Frequent False Detection of Positive Selection by the Likelihood Method with Branch-Site Models’, *Molecular Biology and Evolution*, 21(7), pp. 1332–1339. Available at: https://doi.org/10.1093/molbev/msh117.

Zhang, S. *et al.* (2015) ‘Function and evolution of Magnaporthe oryzae avirulence gene AvrPib responding to the rice blast resistance gene Pib’, *Scientific Reports*, 5, p. 11642. Available at: https://doi.org/10.1038/srep11642.