An integrated approach to enable rapid scalable upstream production of subunit vaccines with *Pichia pastoris* (*Komagataella phaffii*)

by

Andrew M. Biedermann

M.S. Chemical Engineering Practice, Massachusetts Institute of Technology, 2019
B.S. Chemical Engineering, University of Virginia, 2017

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering
at the
Massachusetts Institute of Technology

May 2022

Author _____
Andrew M. Biedermann
Department of Chemical Engineering
May 2022

Certified by _____
J. Christopher Love
Raymond A. (1921) and Helen E. St. Laurent Professor of Chemical Engineering
Thesis Supervisor

Accepted by _____
Patrick S. Doyle
Robert T. Haslam (1911) Professor of Chemical Engineering
Singapore Research Professor
Chairman, Committee for Graduate Students

An integrated approach to enable rapid scalable upstream production of subunit vaccines with *Pichia pastoris* (*Komagataella phaffii*)

by

Andrew M. Biedermann
Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Recent experience with the COVID-19 pandemic motivates the development of vaccine designs and manufacturing technologies tailored to enable widespread access in low- and middle-income countries (LMICs). Due to limited supply, vaccines targeting SARS-CoV-2 were initially distributed primarily to high-income countries. The distribution of vaccines which were eventually made available to LMICs was complicated by logistical challenges such as the need to maintain cold-chain integrity for available vaccine modalities. These technological limitations restricted vaccine access, ultimately driving up excess deaths in LMICs and increasing global pandemic risk by enabling SARS-CoV-2 to rapidly accumulate potentially harmful mutations in an unprotected population. Resolving underlying vaccine supply and distribution challenges for LMICs will be essential to controlling the COVID-19 pandemic and to enable better response to future pandemic threats but will require improved vaccine design and manufacturing.

Subunit vaccines produced with the yeast, *Komagataella phaffii*, could significantly improve access to vaccines in LMICs, owing to their potential for rapid development timelines, high productivity in existing manufacturing capacity, thermostability, and strong efficacy. In this thesis, we present an integrated approach to improve vaccine design and manufacturing in *K. phaffii*. In the first part, we demonstrate that *K. phaffii* strains engineered to eliminate the need for methanol-feeding enable improved production of SARS-CoV-2 RBD antigen. Obviating the need for methanol improved cell health and enabled production of clinical material in a 1200 L bioreactor, larger than would have been possible with traditional methanol-feeding. The benefits of methanol-free engineering appear to be generalizable to other proteins of interest. In the second part, we present a novel "modular blending" approach to media development. This new method enabled the design of a soluble medium with 2x higher productivity than our previous best defined production medium and highlighted the importance of lipid supplementation and carbon metabolism for optimal heterologous protein production in *K. phaffii*. Finally, viral antigens typically require multimeric display to induce strong immune responses, but common nanoparticle display technologies are difficult to produce. We present initial design and experimental work towards the secreted production of novel protein nanoparticles tailored for optimal production in *K. phaffii*.

Thesis Supervisor: J. Christopher Love
Title: Raymond A. and Helen E. St. Laurent Professor of Chemical Engineering

Acknowledgements

I would like to thank my advisor, Chris Love, for his guidance and encouragement throughout my thesis work. Chris helped me become a more organized thinker and clearer communicator and gave me the freedom to pursue and hone a broad range of interests, progressing from perfusion bioreactors, to media development, strain engineering, protein design, and beyond. I would also like to thank my thesis committee members, Kris Prather and Greg Stephanopoulos, for their valuable feedback and mentorship throughout my graduate research. Their questions challenged me to think more deeply about the field, the motivations for my work and research direction. I am also grateful to Kerry Love, who helped me make the transition from computational to experimental work early in my PhD and provided many helpful suggestions and useful feedback throughout my graduate research. I would like to give a special thank you as well to my undergraduate research advisors, Michael Shirts and Geoffrey Geise, who first sparked my interest in scientific research. I would not be here without the responsibility and mentorship that they invested in me throughout my undergrad work at U.Va. and I owe much of the modeling expertise used in our later nanoparticle work to their early guidance.

As evident by the proceeding thesis contents, the pandemic had a significant impact on my thesis work, and I am particularly indebted to many close collaborators in the Love Lab and ULTRA project. In particular, I would like to thank Neil Dalvie, my close graduate mentor and friend, for many stimulating discussions and fruitful collaborations. I would also like to thank Sergio Rodriguez and Laura Crowell, who burned the midnight oil to run the InSCyT systems with me during the early days of lockdown and worked with me on many other projects throughout grad school, as well as Danielle Camp, our lab's program manager, for her support and direction throughout the years.

I had the opportunity to work with and mentor talented undergrad and grad students at MIT and I want to thank them for their hard work and great insights. Isabella Gengaro worked closely with me for several years and was integral to early planning of the media development work as well as other exploratory projects in codon optimization and microbiology. I have no doubt she will go on to do excellent research as she begins her own graduate studies next year. I would also like to thank Shuting Shi, who worked closely with me on nanoparticle development and will be continuing our earlier work in her graduate studies. I am excited to see where the science can go in her capable hands.

I would like to thank my many friends and family, who shared many of the highs and lows of this processes with me. I want to thank my parents, Jeff and Ellen, and grandma, Carol, who got me interested in math and science at an early age and helped me develop many of my interests, as well as my siblings, Garrett, Kristen, and Evan. I want to thank Lucky and Charles, who've been close friends since we were first year office mates hurdling homework together; Sid, for our enjoyable discussions ranging from biotech to spelunking and many QGs; and Kara, Jennifer, and the rest of the

practice school squad, for many two-dollar days. I would also like to thank friends from the MIT Biotech Group, whose interests and insights have been truly inspiring.

Finally, I want to thank Linda, for her close friendship, support, and humor throughout this process. You've celebrated each success with me, cheered me up during setbacks, and listened patiently to excited rants about protein soccer balls. I've been so lucky to share this experience with you and am looking forward to our next steps together.

Table of Contents

List of figures and tables

1.Introduction

1.1 Motivating vaccine design and manufacturing tailored for unmet need in low- and middle-income countries

The COVID-19 pandemic has emphasized the need for vaccine design and manufacturing tailored to meet demand in low- and middle-income countries (LMICs). As shown in **Figure 1.1**, the initial distribution of vaccines to combat the SARS-CoV-2 pandemic has been disproportionately skewed toward high-income countries. This has led to a large increase in excess deaths in LMICs and provided a refuge for SARS-CoV-2 in unvaccinated populations (Kim, Marks, & Clemens, 2021). In addition to the harm done to people in LMICs, such a refuge also allows the virus to accumulate further mutations, increasing global risk by potentiating the evolution of more infectious strains or strains which may evade the protective immunity induced by prior vaccinations (Pulliam et al., 2021).

COVID-19 vaccine doses administered vs GDP per capita, Sep 23, 2021

For vaccines that require multiple doses, each individual dose is counted. As the same person may receive more than one dose, the number of doses per 100 people can be higher than 100. GDP per capita is adjusted for price differences between countries (it is expressed in international dollars).

Source: Official data collated by Our World in Data – Last updated 24 September 2021, 14:00 (London time), World Bank

OurWorldInData.org/coronavirus • CC BY
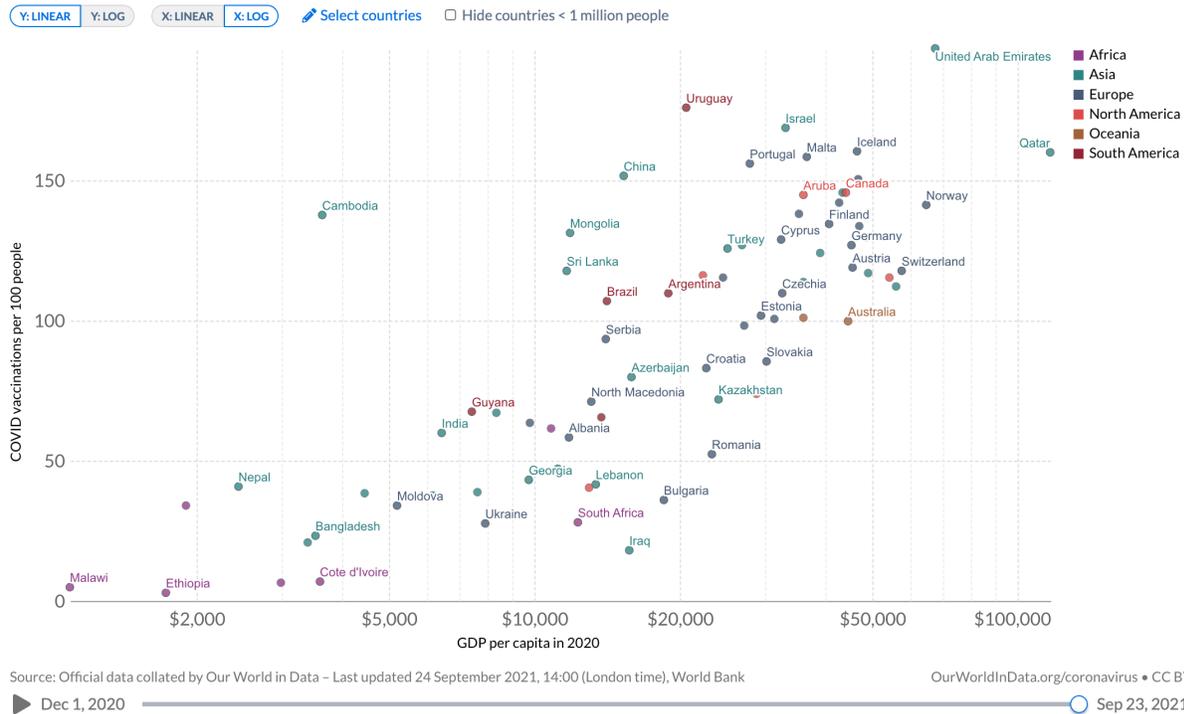
Figure 1.1 Access to SARS-CoV-2 vaccines as of September of 2021 was highly correlated with national GDP per capita

(Sourced from database of Mathieu et al., 2021)

Insufficient vaccine supply and challenges in vaccine distribution limited access to COVID-19 vaccines in LMICs. In 2021, vaccine demand far exceeded vaccine supply, due to limited manufacturing capacity and the insufficient volumetric productivities achieved by commercially available vaccines (Kim et al., 2021; Sell et al., 2021). Recognizing supply limitations, high-income countries used advanced purchase agreements to guarantee early access to vaccines at high prices (Duke Global Health Innovation Center, 2021). Organizations, such as COVAX, also attempted to secure advanced purchase agreements on behalf of LMICs; however, many countries

restricted vaccine exports from domestic manufacturers, limiting distribution of vaccines to many LMICs without existing biomanufacturing capacity (Massinga Loembé & Nkengasong, 2021). Challenges meeting and maintaining cold-chain integrity during vaccine distribution further limited vaccine access in LMICs, especially for temperature-sensitive mRNA vaccines (Acharya, Ghimire, & Subramanya, 2021; Massinga Loembé & Nkengasong, 2021).

The factors that limited supply and complicated vaccine distribution must be addressed, both for long-term management of SARS-CoV-2 variants and in preparation for future pandemic pathogens. Building additional manufacturing capacity and installing or upgrading existing distribution infrastructure could take years and cost billions of dollars (Coffman et al., 2021). Alternatively, improvements to underlying vaccine design and manufacturing could also address these challenges by resolving limitations and enabling more efficient use of existing resources. This motivates exploration and development of additional vaccine modalities and manufacturing technologies tailored to meet the needs of LMICs.

1.2 Motivating development of subunit vaccines for LMICs

An ideal vaccine modality tailored to improve access in LMICs would need to be capable of rapid development, strong efficacy with minimal side effects, high volumetric productivity with significant existing manufacturing capacity, and compatible with minimal cold-chain requirements. There are many vaccine modalities, and each modality has trade-offs which impact its utility for use in LMICs. Thus, careful consideration must be given to selection of an ideal vaccine modality.

As shown in **Figure 1.2**, vaccines can be broken into four general classes:

Nucleic acid, viral, vector-based, and subunit vaccines (a.k.a. protein-based vaccines).

Nucleic acid vaccines, such as the Pfizer and Moderna vaccines, can be rapidly

developed and can achieve high efficacy; however, there is limited existing

manufacturing capacity for nucleic acid therapies, they can cause inflammatory side

effects that may exacerbate vaccine hesitancy, and they may require cold-chain storage

at temperatures as low as -80 $^{\circ}$C due to the poor stability of mRNA (Baden et al., 2021;

Sharma, Sultan, Ding, & Triggle, 2020). Inactivated viral vaccines can achieve good

efficacy, but also may have inflammatory side effects, can present potential safety

issues during manufacturing due to the cultivation of live pathogen, and rely on

cultivation in slow-growing mammalian cell lines, resulting in slower development and

manufacturing timelines (Haque & Pant, 2020). Vector-based vaccines use adenovirus

vectors produced in HEK cells to deliver DNA encoding the SARS-CoV-2 spike protein

into cells (Kallel & Kamen, 2015). While this enables a platform manufacturing approach

and avoids safety issues associated with culturing live pathogen, adenovirus-based

vaccine efficacy may decrease upon re-administration due to immunogenic responses

to the original vector, so each vector may have limited utility as a vaccine booster or for

development of subsequent vaccines against other pathogens (S. Ahi, S. Bangari, & K.

Mittal, 2011).
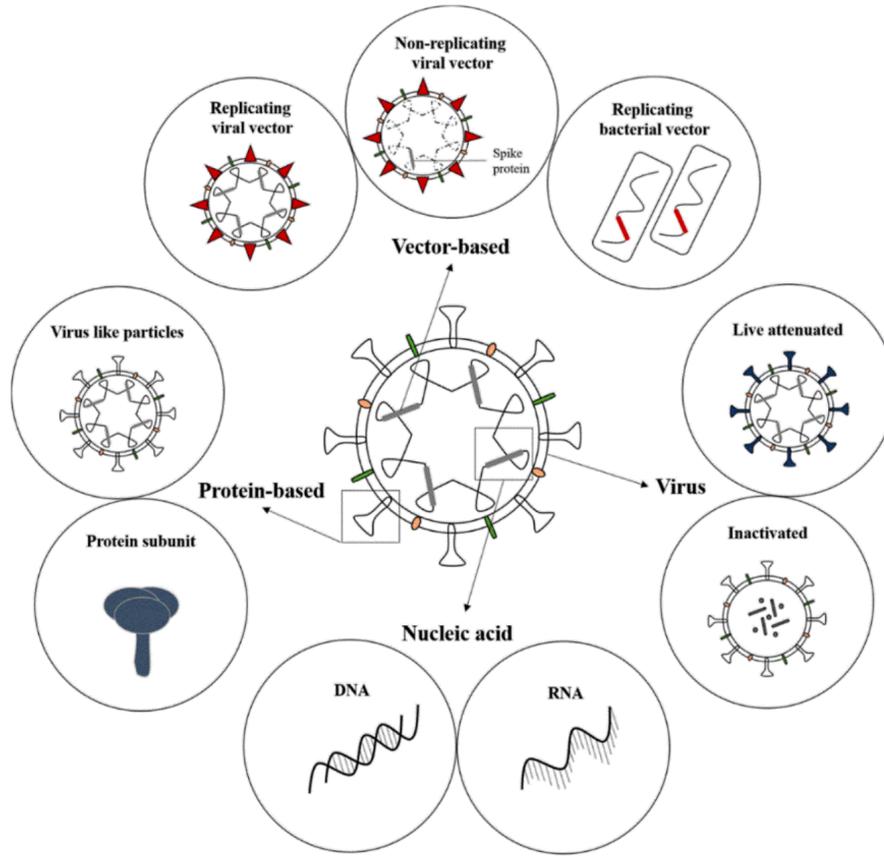
Figure 1.2 Overview of common vaccine modalities

(Sourced from Mellet & Pepper, 2021)

Among available vaccine modalities, subunit vaccines (a.k.a. protein-based vaccines) may have the most potential to improve vaccine access in LMICs. Subunit vaccines can achieve high efficacy with few side effects, demonstrate high thermal-stability, and be manufactured at large-scale using existing capacity—although further improvements to subunit vaccine design and manufacturing technologies are needed to realize these potential benefits within a single vaccine candidate. Novavax's COVID-19 vaccine candidate demonstrated best-in-class efficacy with minimal side effects in Ph3 study; however, manufacturing delays extended its development timeline (Heath et al.,

2021; Tinari & Riva, 2021). Icosavax leverages computational design to develop highly

thermal stable synthetic viral-like particle (VLP) subunit vaccines, observing minimal

degradation at temperatures up to 80 °C for early nanoparticle designs, such as the I3-

01 nanoparticle (Xu et al., 2016). Separately, other prominent subunit vaccines, such as

the Shanvac-B hepatitis B vaccine, have been manufactured in microbial hosts which

grow 6-8x faster than mammalian cell lines, enabling rapid development and large-scale

high volumetric productivity manufacturing in existing facilities (Ghosh, 2017; Matthews,

Wright, et al., 2017).

1.3 Motivating the use of *K. phaffii* for rapid, scalable subunit vaccine production

The use of a microbial host is particularly important for improving access to

subunit vaccines in LMICs, due to its impact on development speed, productivity, and

available manufacturing capacity (Brady & Love, 2021). Mammalian cell lines can be

used to secrete high titers of complex proteins but grow slowly (doubling time of ~1 day)

on expensive media to low cell densities and produce a high host cell protein burden,

complicating downstream purification. In a pandemic setting, mammalian host

manufacturing capacity may also be strained by therapeutic monoclonal antibody

development, which often relies on expression in a mammalian host (Kelley, 2020).

Bacteria, such as *E. coli*, grow quickly on cheap media (doubling time of ~20 minutes) to

high cell densities, but lack the necessary organelles to fold, glycosylate, and secrete

complex proteins, requiring lysis and complex downstream processes to recover

heterologous proteins. Thus, while production via *E. coli* may have significant available

manufacturing capacity, *E. coli* may require more bioprocessing steps and could be limited in the range of proteins they may produce at high titers. Yeasts, such as *K. phaffii* (known commonly as *Pichia pastoris*), can grow rapidly to high cell density (doubling time ~3 hrs, 8x faster than mammalian cells) and can secrete more complicated proteins with minimal host cell protein burden, leading to simpler downstream purification (Matthews, Wright, et al., 2017). They are also a proven host for vaccine manufacturing, with extensive capacity available from manufacturers such as the Serum Institute of India and Biological E (Ghosh, 2017; Kleanthous et al., 2021).

Due to these advantages, there is significant interest in developing improved vaccine designs and manufacturing approaches for the expression of vaccine candidates in *K. phaffii*. In an ideal form, such a technological platform could potentially deliver safe and effective vaccines in timelines competitive with nucleic acid and viral vector-based modalities, manufactured at microbial scales (50,000 L+) with low costs and no cold-chain requirements.

1.4 Thesis objectives

Recognizing the potential of *K. phaffi*-based bioprocesses to enable rapid, scalable production of effective subunit vaccines, this thesis aims to develop broadly applicable strategies for improved upstream manufacturing of subunit vaccines with *K. phaffii*. Effective subunit vaccines typically consist of two parts: Antigens derived from the target pathogen and a presentation platform which enables immunogenic display of pathogen-derived antigens. This thesis work presents an integrated approach to enable improved production of both components.

Bioprocessing strategies to improve production of pathogen-derived antigens

While often beneficial when used judiciously, mutating antigens to improve manufacturability could remove important immunological epitopes or induce unwanted modification-specific responses. There is therefore value in developing strain and process improvements which enable improved production of pathogen derived antigens without sequence modifications.

Chapter 2: Scalable, methanol-free manufacturing of the SARS-CoV-2 receptor binding domain in engineered *Komagataella phaffii*. *K. phaffii* enables rapid biomass accumulation and strong expression of heterologous protein under the control of the pAOX1 promoter; however, the use of methanol as a carbon source during protein production results in the generation of harmful metabolic byproducts such as formaldehyde and hydrogen peroxide, high heat and oxygen demands due to the stoichiometry of methanol metabolism, and is incompatible with many potential manufacturing facilities due to the safety and specialized storage requirements which result from the large-scale use of methanol. In this chapter, we engineer strains to enable methanol-free activation of the pAOX1 promoter and demonstrate that such strains enable improved productivity and cell health during production of a SARS-CoV-2 RBD subunit vaccine component. Following initial strain characterization and pilot scale experiments in our benchtop perfusion reactor, InSCyT, our methanol-free RBD expressing strains were transferred to the Serum Institute of India, where they enabled production in a 1200 L bioreactor to support early clinical trials of a COVID-19 vaccine candidate. Product quality was consistent between lab scale shake flask experiments

and a 1200 L bioreactor run, demonstrating good scaling behavior. This production

strategy also enables strong production of RBD subunit components modified to match

other SARS-CoV-2 variants of concern. The Serum Institute also requested that future

strains contain methanol-free genetic edits, providing further validation of the initial

motivation for this work.

Chapter 3: Optimization of methanol-free production of a rotavirus-derived subunit

vaccine antigen using multiplex genetic engineering. Following our early success using

strains engineered to enable methanol-free activation of pAOX1 to produce RBD, we

next considered whether similar methanol-free expression strategies could be

generalizable for the expression of other heterologous proteins of interest. As a test

case, we attempted to achieve methanol-free expression of a rotavirus-derived subunit

vaccine component, P[8], comparable to a methanol-fed WT control. To our surprise,

however, we found that overexpression of additional transcription factors associated

with pAOX1 induction was required to achieve comparable expression of P[8]. While the

exact degree of engineering required may vary somewhat from protein-to-protein, our

results in this chapter demonstrate methanol-free expression comparable to or better

than a methanol-fed WT control for 4 additional model proteins, suggesting that strong

methanol-free production can be consistently achieved in *K. phaffii* cultivations.

Chapter 4: Modular development enables rapid design of media for alternative hosts.

Media optimization is an important step in process optimization for any protein of

interest; however, despite its ubiquity, methods for media development are often labor

intensive, low throughput, or rely on extensive analytical capabilities. In this chapter, we

address these challenges by developing a novel modular media blending methodology

that enable rapid and routine optimization of batch culture media for any measurable phenotype of interest. Our method enabled the creation of a heterologous protein production medium that achieved a 2x improvement in secreted titer of P[8], relative to our previous best defined media formulations. The resulting data set emphasizes the importance of lipid supplementation and carbon feeding strategy for optimal production of heterologous protein in *K. phaffii*, and the methodology can be easily adapted to improve media designs for other alternative hosts of interest.

Tailoring display platform design for optimal production in *K. phaffii*

As monomers, small proteins derived from a viral capsid are not naturally arrayed in regular intervals to mimic presentation in a viral capsid. This leads to poor clustering of B-cell receptors and attenuated immunogenicity in many cases, leading to lower efficacy and limiting the utility of small monomeric subunit vaccines (Kelly, Kent, & Wheatley, 2019; Veneziano et al., 2020). Recently, novel nanoparticle platforms have been developed which substantially improve immunogenicity; however, none of these platforms is designed explicitly for rapid, scalable bioprocessing, leaving significant unrealized bioprocessing efficiency gains. Early work outlines potential bioprocessing efficiency gains and provides a roadmap toward their realization.

Chapter 5: Initial progress on the computational design of an ultra-low cost protein nanoparticle vaccine platform for immunogenic display of viral antigens. By definition, antigen display technologies should not contain important immunological epitopes, enabling considerably greater freedom for protein design. In this chapter, we develop initial designs for a nanoparticle vaccine platform tailored explicitly for optimal

production by secretion in a *K. phaffii* co-culture. Secreted titer tends to decrease with increasing protein size, due to the greater folding and assembling complexity. We therefore designed two-component protein nanoparticles to be secreted as separate subunits that assemble in the supernatant. Initial work to redesign a previously reported nanoparticle component out of the Institute for Protein Design resulted in a significant improvement in secreted titer, providing early experimental proof-of-concept. We then developed new computational designs that may simplify the protein expression challenge further, by avoiding intracellular homo-multimerization of individual components. The resulting work provides an early update on and roadmap for ongoing work, as well as insights into relevant technical challenges and solutions encountered while initiating Rosetta-based protein design efforts in our lab, which may be of interest to other scientists interested in entering this space.

2. Scalable, methanol-free manufacturing of the SARS-CoV-2 receptor binding domain in engineered *Komagataella phaffii*

2.1 Motivation

As new variants of SARS-CoV-2 emerge, continued development of diagnostics,

vaccines, and reagents remains essential to address the COVID-19 pandemic. The

SARS-CoV-2 spike protein is an essential reagent for serological assays, and a

component of several protein-based vaccines (Guebre-Xabier et al., 2020; Tian et al.,

2020). Vaccine candidates based on protein subunits are also important ones for

enabling interventions for the pandemic in low- and middle-income countries (LMICs)

due to existing large-scale manufacturing facilities and less stringent temperature and

storage requirements for distribution (Dai et al., 2020). We and others have reported

vaccine designs based on the receptor binding domain (RBD) of the spike protein

(Dalvie et al., 2021). In these designs, the RBD can be produced independently, and

subsequently displayed on protein or lipid nanoparticles for enhanced immunogenicity

(Cohen et al., 2021; Walls et al., 2020). The 201 amino acid RBD is an especially

promising antigen for accessible vaccines because it can be manufactured at low cost

and high volumes in microbial hosts (Chen, Hotez, & Bottazzi, 2020; Pollet et al., 2020).

Here, we report an engineered yeast strain with enhanced secretion of the SARS-CoV-2

RBD from the circulating variants of Wuhan Hu-1, B.1.1.7, and B.1.351 strains of the

virus. This engineered host has been successfully deployed at 1,200 L scale to produce

a vaccine component currently in clinical trials.

The methylotrophic yeast *Komagataella phaffii* (*Pichia pastoris*) is routinely used

for the production of therapeutic proteins at large volumes because of its high-capacity

eukaryotic secretory pathway (K. R. Love, Dalvie, & Love, 2018). Another key

advantage of this production host is the strong, tightly regulated, methanol-inducible

promoter, P_{AOX1}, used for expression of the recombinant gene (Ahmad, Hirz, Pichler, &

Schwab, 2014). This promoter enables outgrowth to high cell densities with inexpensive

feedstock like glycerol before induction of the recombinant gene with methanol feed.

Methanol can pose challenges, however, in large-scale facilities, including high heat

generation during fermentation and flammability concerns while in storage (Potvin,

Ahmad, & Zhang, 2012). The impact of these challenges is that facilities require specific

designs or modifications to handle methanol. This requirement could limit the number of

manufacturing facilities available for the production of vaccine components like the RBD

antigens in *K. phaffii* in a pandemic. We sought to reduce or eliminate the requirement

for methanol for efficient secretion of the RBD.

2.2 Results and Discussion

We previously reported the production of the SARS-CoV-2 RBD (Wuhan-Hu-1

sequence) in an engineered variant of *K. phaffii* (Brady et al., 2020; Dalvie et al., 2021).

To assess the feasibility of methanol-free production, we cultivated the strain expressing RBD regulated under the native AOX1 promoter, and induced expression of the recombinant gene with varying amounts of methanol (**Figure 2.1A**). Interestingly, the approximate secreted titers of RBD increased as the concentrations of methanol were reduced. We also induced protein production with a combination of methanol and sorbitol—a supplementary carbon source that does not repress P_{AOX1} expression—and observed a further increase in titer.
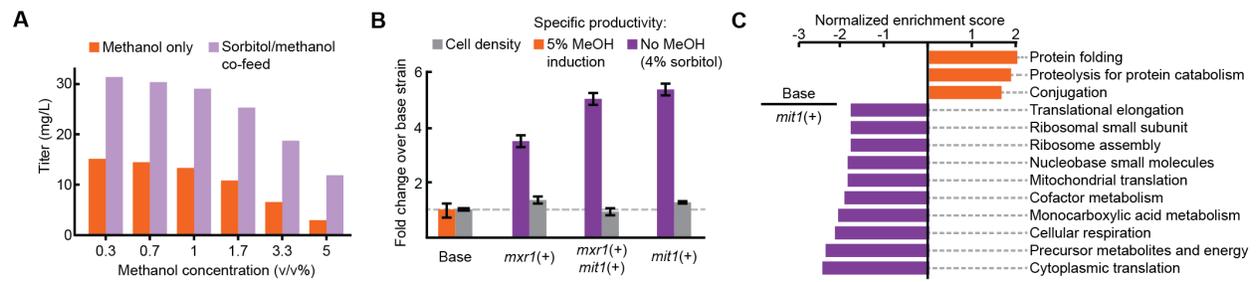
Figure 2.1 Improved productivity and decreased stress in methanol-free RBD expression

(A) Approximate titers of secreted RBD from individual cultures of the base strain in 3 mL plate culture, measured by reverse-phase liquid chromatography. (B) Performance of three engineered strains in 3 mL plate culture. Error bars represent standard deviation of three biological replicates. (C) Enriched gene sets between the base strain (orange) and the *mit1+* strain (purple).

Given our observation of improved productivity with reduced quantities of methanol in batch cultivations, we hypothesized that we could achieve or maintain productive secretion of the RBD in the absence of methanol with suitable engineering of the strain. Expression of genes regulated by P_{AOX1} in wild type *K. phaffii* in the absence

of methanol is inconsistent, even with non-repressive carbon sources like sorbitol (Vogl, Sturmberger, et al., 2018). Several studies, however, have demonstrated that constitutive overexpression of activating transcription factors like *mit1* and *mxr1* can lead to consistent activation of P_{AOX1} without methanol (Shi et al., 2019; Vogl, Sturmberger, et al., 2018). To test production of RBD without methanol, we integrated additional copies of the endogenous transcription factors *mit1* and *mxr1* into the *K. phaffii* genome under a glycerol-repressible promoter (Dalvie et al., 2019). We cultivated these strains for protein production by feeding with only sorbitol (**Figure 2.1B**). We observed a >3-fold increase in specific productivity in all strains, particularly with a strain containing only one extra copy of the transcription factor *mit1* (>5-fold).

       To assess the potential source of improved productivity, we performed a comparison of the methanol-fed base strain and the modified, sorbitol-fed *mit1+* strain. We observed no intracellular accumulation of RBD protein in either strain (**Figure A.1A**). Next, we examined the transcriptomes of the methanol-fed initial strain and the modified, sorbitol-fed *mit1+* strain by RNA-sequencing. The sorbitol-fed *mit1+* strain appeared to produce less RBD transcript than the methanol-fed base strain, but the difference was not significant (unpaired t-test, p=0.06) (**Figure A.1B**). We analyzed the variations in gene expression by gene set enrichment analysis (**Figure 2.1C**). We observed significantly higher expression of genes associated with protein folding stress in the methanol-fed condition compared to the sorbitol-fed *mit1+* condition (family-wise error p=0.003). These results suggest that sorbitol-fed *mit1+* may improve productivity by mitigating protein folding stress associated with RBD production.

To determine whether the observed reduction in protein folding stress was due to the sorbitol feed or the *mit1+* engineering, we cultivated the *mit1+* strain with different feed conditions (**Figure A.1C**). We observed that the specific productivity of secreted RBD was reduced in 5% methanol feed, even with the *mit1+* engineering. The improvement in specific productivity, therefore, can be primarily attributed to elimination of methanol as a carbon source. This observation is consistent with previous transcriptomic studies about methanol metabolism in *K. phaffii* (Lin, He, Xu, & Yu, 2021; Vanz et al., 2012). Further studies are warranted to determine the interplay between the transcript level and types and quantities of carbon source on productivity.

After comparing the specific productivity of the methanol-free strain (*mit1+*) to the methanol-induced (base) strain, we assessed the production of RBD using both strains on InSCyT, a continuous, automated, perfusion-based manufacturing platform (Crowell et al., 2018). The base strain exhibited low titers (~30 mg/L) in perfusates and significant cell lysis after ~120 h of fermentation in perfusion (**Figures 2.2A&B**). In contrast, the *mit1+* strain maintained protein secretion at >50 mg/L/day for the duration of a >200 h campaign. RBD purified from the perfusates produced by the base strain also contained more host-related impurities than RBD from the *mit1+* campaign (**Figure 2.2C**). These results from the sustained production of RBD, including the cell lysis observed in the base strain, are consistent with the observations for increased cellular stress relative to the *mit1+* strain, and suggest the transcriptional changes observed also translated into variation in protein expression as well.
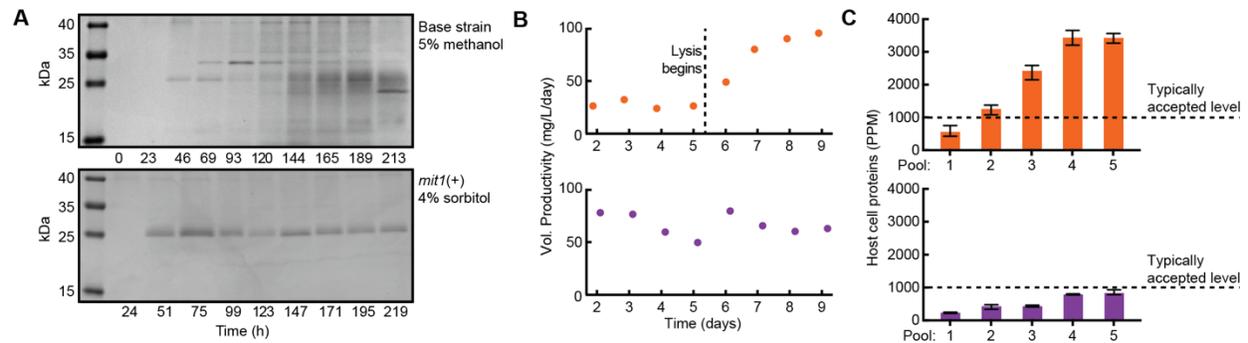
Figure 2.2 Sustained productivity of the methanol-free strain in perfusion fermentation
(A) Reduced SDS-PAGE of upstream reactor samples for the duration of each
campaign. (B) Upstream reactor titer of RBD. (C) Host cell protein concentrations in
purified pools of RBD, measured by ELISA. Error bars represent standard deviation of
three technical replicates.

From these data for the improved production of RBD in bioreactors with the
modified strain without methanol, we then generated an *mit1+* strain that expressed
RBD with a C-terminal fusion of SpyTag, a short peptide that can mediate a
transpeptidation reaction with a cognate SpyCatcher polypeptide, which can be
presented on protein nanoparticles for example (Reddington & Howarth, 2015). We
expressed and purified the RBD-Spytag from this strain in a 200 mL shake flask culture.
We also transferred this *mit1+* strain encoding RBD-SpyTag, to a facility for GMP
manufacturing in a 1,200 L fed batch process. In this process, the strain produced 21
mg per liter of fermentation of purified, clinical quality RBD-SpyTag, or approximately >1
million doses from a single reactor batch, assuming a vaccine formulation with 25 µg of
RBD-SpyTag per dose. The two purified products from each production scale were
similar by SDS-PAGE, and exhibit nearly identical glycan profiles, indicating consistency

in the quality attributes of the molecules produced at these two scales with this modified

strain for methanol-free production (**Figure 2.3**).
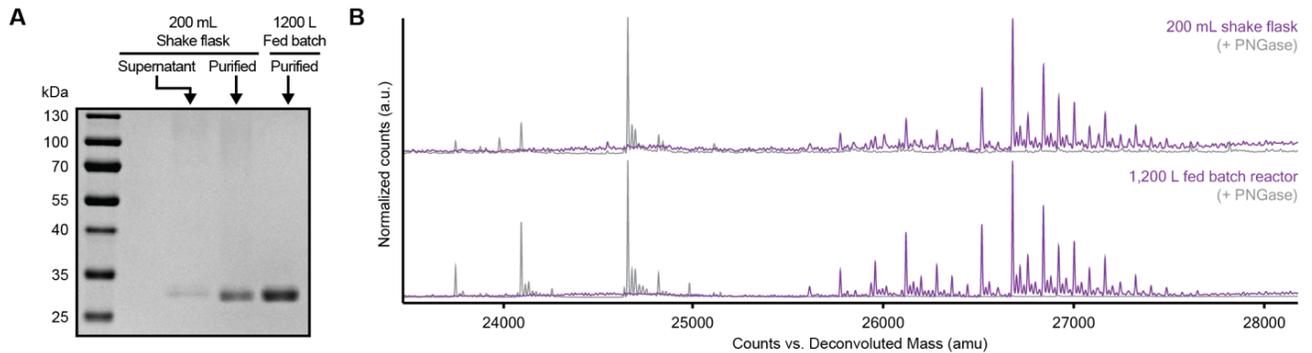
Figure 2.3 RBD-SpyTag produced at lab scale and GMP scale

A) Reduced SDS-PAGE of RBD-SpyTag in crude shake flask supernatant, purified from

shake flask cultivation, and purified from a fed batch process. B) Intact mass spectra of

purified RBD-SpyTag from each manufacturing process. Overlayed spectra are before

and after treatment with PNGase.

We next sought to assess whether or not this modified *mit1+* strain could improve

the production of sequence variants for other circulating SARS-CoV-2 virus strains as

well. We generated strains expressing RBD-B.1.1.7 and RBD-B.1.351 in both the base

and *mit1+* strain backgrounds, and evaluated their specific productivities in different

media for production (**Figure 2.4**). In all strains, reduced methanol feed improved

productivity. For all RBD variants, only *mit1+* engineered strains maintained improved

productivity in the absence of methanol. This result demonstrates that the engineered

mit1+ strain could facilitate new cell lines for manufacturing other RBD variants without

methanol for seasonal vaccine boosters or next-generation vaccine candidates for emerging variants.
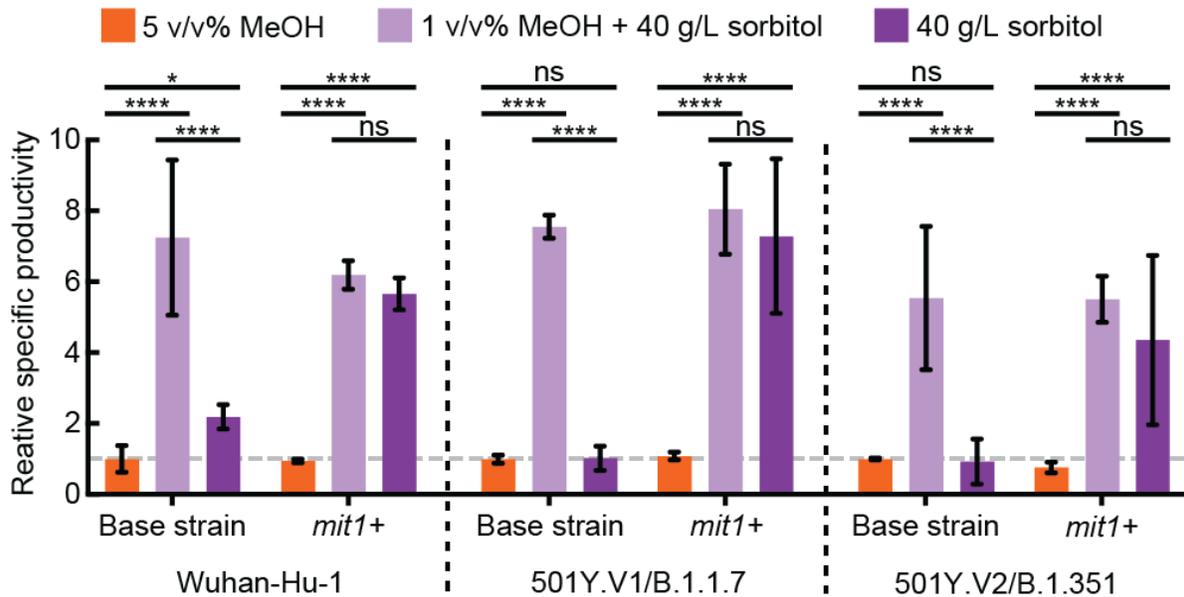
Figure 2.4 Methanol-free production of RBD variants in 3 mL culture
Error bars represent standard deviation across three biological replicates. Significance was determined by multiple t-tests with Holm Sidak correction. $*p<0.01$, $****p<0.000001$.

In conclusion, we report here a strain that enables the manufacturing of SARS-CoV-2 RBD variants without methanol. This strain exhibits improved secreted productivity due to a reduction in protein folding stress. We demonstrated sustained productivity from the strain in a perfusion process, and scale-up to a large-scale, methanol-free fed batch process to produce a vaccine component currently in clinical trials. In this case, manufacturing at the 1,200 L scale was possible with the elimination of the requirement for methanol in the medium. Strains engineered for use without methanol and increased productivity could facilitate manufacturing of RBD and other

antigens for vaccine candidates at large volumes and low costs to enable accessible

and affordable vaccines for global use.

2.3 Materials and methods

2.3.1 Yeast strains

All strains were derived from wild-type *Komagataella phaffii* (NRRL Y-11430).

The base strain was described previously (Brady et al., 2020). The gene containing the

RBD was codon optimized, synthesized (Integrated DNA Technologies), and cloned into

a custom vector. The RBD vector was transformed as described previously (Dalvie et

al., 2019). Transcription factors *mit1* and *mxr1* were integrated into the genome near

genomic loci GQ67_02967 and GQ67_04576, respectively, using a markerless

CRISPR-Cas9 system described previously (Dalvie et al., 2019). Both *mit1* and *mxr1*

were under control of the P_{CAT1} promoter from *K. phaffii*. Sequences for P_{CAT1}, *mit1*, and

mxr1 were amplified from the *K. phaffii* genome.

2.3.2 Cultivations

Strains for initial characterization and titer measurement were grown in 3 mL

culture in 24-well deep well plates (25°C, 600 rpm), and strains for protein purification

were grown in 200 mL culture in 1 L shake flasks (25°C, 250 rpm). Cells were cultivated

in Rich Defined Media, described previously (Matthews, Kuo, Love, & Love, 2017a).

Cells were inoculated at 0.1 OD600, outgrown for 24 h with 4% glycerol feed, pelleted,

and resuspended in fresh media with methanol or sorbitol feed to induce recombinant

gene expression. Supernatant samples were collected after 24 h of production, filtered,

and analyzed. InSCyT bioreactors and purification modules were operated as described previously (Crowell et al., 2018; Dalvie et al., 2021).

2.3.3 Analytical assays for protein characterization

Purified protein concentrations were determined by absorbance at 280 nm. SDS-PAGE was carried out as described previously (Crowell et al., 2018). Supernatant titers were measured by reverse phase liquid chromatography as described previously (Dalvie et al., 2021), and normalized by cell density, measured by OD600. Intact mass spectrometry was performed as described previously (Dalvie et al., 2021) but with the following modifications: LC gradient of 5-95% solvent B over 4 minutes at a flow rate of 0.8mL/min, and 250 V fragmentor voltage.

2.3.4 Transcriptome analysis

Cell were harvested after 18 h of production at 3 mL plate scale. RNA was extracted and purified according to the Qiagen RNeasy kit (cat #74104) and RNA quality was analyzed to ensure RNA Quality Number >6.5. RNA libraries were prepared using the 3'DGE method and sequenced on an Illumina Miseq to generate paired reads of 20 (read 1) and 72 bp (read 2). Sequenced mRNA transcripts were demultiplexed using sample barcodes and PCR duplicates were removed by selecting one sequence read per Unique Molecular Identifier (UMI) using a custom Python script. Transcripts were quantified with Salmon version 1.1.0 (Patro, Duggal, Love, Irizarry, & Kingsford, 2017) and selective alignment using a target consisting of the *K. phaffii* transcripts, the RBD, and selectable marker transgene sequences and the *K. Phaffii* genome as a

selective alignment decoy. Expression values were summarized with tximport version

1.12.3 (Soneson, Love, & Robinson, 2016) and edgeR version 3.26.8 (McCarthy, Chen,

& Smyth, 2012; Robinson, McCarthy, & Smyth, 2009). Expression was visualized using

$log_2(Counts per Million + 1)$ values. Gene set enrichment analysis (GSEA) was

performed with GSEA 4.1.0 using Wald statistics calculated by DESeq2 (M. I. Love,

Huber, & Anders, 2014) and gene sets from yeast GO Slim (Subramanian et al., 2005).

RNA sequencing data is available in the NCBI Gene Expression Omnibus, accession

GSE183408.

3. Optimization of methanol-free production of a rotavirus-derived subunit vaccine antigen using multiplex genetic engineering

This work is based on a manuscript in preparation to be submitted to *Biotechnology and Bioengineering*. Biedermann, A. M., Dalvie, N. C., Gengaro, I. R., Lorgeree, T., Love, K. R. & Love, J. C. Assessing the impact of multiplex engineering on methanol-free pAOX1 activation and protein expression in *Komagataella phaffii*. *Prep.* (2022).

3.1 Motivation

Komagataella phaffii is a widely used host for heterologous protein expression, owing to its low HCP burden and a strong methanol-inducible promoter, pAOX1 (Matthews, Wright, et al., 2017). The inducibility of genes controlled by the promoter pAOX1 in the presence of methanol allows fermentations in which rapid outgrowth to high cell density is achieved on a repressive carbon source, such as glycerol, and subsequent expression of a heterologous transgene when using a feedstock containing methanol. This capability of pAOX1 to separate stages of growth and production is advantageous, but the use of methanol as the chemical inducer has two significant disadvantages for large-scale bioprocesses. First, methanol presents significant risks for flammability and toxicity, requiring specialized on-site storage facilities (Potvin et al., 2012). Second, the metabolism of methanol requires ~30% higher heat and oxygen transfer rates per C-mol than other common sources of carbon, and results in the formation of toxic metabolic byproducts, such as formaldehyde and hydrogen peroxide. These by-products can adversely affect cell culture health and performance, especially

in batch or fed-batch processes (Hartner & Glieder, 2006; P. Zhang et al., 2010). These two limitations can limit the suitability of processes for existing facilities, and consequently the available manufacturing capacity for products of interest, such as SARS-CoV-2 vaccine candidates (Dalvie, Biedermann et al., 2021). Methanol-free bioprocesses are also required for some product-specific applications, such as the production of soy leghemoglobin for Impossible Burgers (Fraser, Shitut, Agrawal, Mendes, & Klapholz, 2018).

One approach to mitigate the challenges associated with using methanol has been to create strains that overexpress certain transcription factors (TFs) to facilitate methanol-free induction of pAOX1: There are several examples wherein one of three known activating TFs (MXR1, MIT1, and PRM1) are overexpressed (Lin-Cereghino et al., 2006; Sahu, Krishna Rao, & Rangarajan, 2014; X. Wang et al., 2016). A two-stage feeding strategy in plate-scale cultivations using glucose followed by glycerol achieved 77% of methanol-fed WT GFP levels by overexpressing Mit1 and knocking out *Mig1*, *Mig2*, and *Nrg1* (J. Wang et al., 2017). Using an innovative pCAT1-driven de-repressible expression system to test the impact of overexpressing Mit1, Mxr1, and Prm1 individually, methanol-free expression for dTom, eGFP, HRP, and DtHNL was attenuated relative to expression with methanol, although one protein (CalB) did exhibit enhanced production with this system (Vogl, Glieder, et al., 2018). Independently, strains with overexpressed Mxr1 have also shown potential for methanol-free expression of GFP (Chang, Hsiung, Hong, & Huang, 2018). These examples all highlight the potential for engineering the host to favor activation of genes in the absence of methanol. The general productivities of strains engineered with these

overexpressed TFs, however, have not shown to date production comparable to, or exceeding, what is feasible by including methanol in the feedstocks for fermentation for many proteins of interest.

The examples reported to date have largely focused on overexpression of individual activating TFs. It remains unclear whether synergistic effects on gene activation using pAOX1 may emerge with a multiplicity of altered profiles of genes, or whether modulation of other correlated genes or pathways could further augment these approaches. Further transcriptional characterization and strain engineering could help inform strategies to consistently create methanol-free bioprocesses with productivity comparable to methanol-fed WT expression systems.

Here, we have used multiplexed strain engineering and RNA sequencing to achieve methanol-free expression comparable to a WT control in engineered strains of *K. phaffii*. Initial attempts to produce a component of a rotavirus-derived subunit vaccine candidate (P[8]) resulted in methanol-free productivity that was lower than a methanol-fed WT control—although favorable results were obtained for this strain with other proteins of interest. To address this special case, we used RNA sequencing to examine the transcriptional changes induced by methanol in a WT P[8]-expressing strain. By cross-referencing known TFs in *K. phaffii* with expression of MUT pathway genes, we discovered a novel methanol-inducible TF, GQ67_00201, which yielded improved pAOX1 activation, and identified and conducted initial characterization of other relevant TFs and MUT pathway genes of interest. The resulting strains achieved significantly improved productivity of P[8]. The resulting strains and data from this work build on prior

studies and demonstrate that *K. phaffii* strains can be engineered to consistently

achieve expression comparable to a methanol-fed WT control.

3.2 Results

3.2.1 P[8] production appears limited by attenuated transcription under methanol-free conditions

Methanol-free expression systems have proven useful for the production of

SARS-CoV-2 receptor binding domain (RBD) antigens (Dalvie, Biedermann et al.,

2021). We wondered whether a similar strategy of overexpressing pAOX1-activating

TFs could enable comparable expression of another rotavirus-derived subunit vaccine

component of interest, P[8]. Using strains from our previous study, we began by

evaluating the methanol-free productivity of P[8] in strains overexpressing either MXR1

or both MXR1+MIT1; however, in contrast to previously reported results for RBD, we

found that neither strain achieved methanol-free productivity comparable to the

methanol-fed WT control (**Figure 3.1A**). Three transcription factors, MXR1, MIT1, and

PRM1, have been shown to bind the pAOX1 promoter (X. Wang et al., 2016), so we

used our host-informed CRISPR toolset to introduce PRM1 to the dual strain, creating a

strain overexpressing all three activating TFs (MXR1+MIT1+PRM1) under the control of

the pCAT1 promoter (Dalvie, Leal, et al., 2020; Vogl, Glieder, et al., 2018). To our

surprise, however, this strain still did not achieve methanol-free productivity of P[8]

comparable to a methanol-fed WT control (**Figure 3.1A**). This issue appeared to be

somewhat specific to methanol-free production of P[8], as the triple activator strain

achieved comparable production of other heterologous proteins of interest (**Figure 3.1B**).
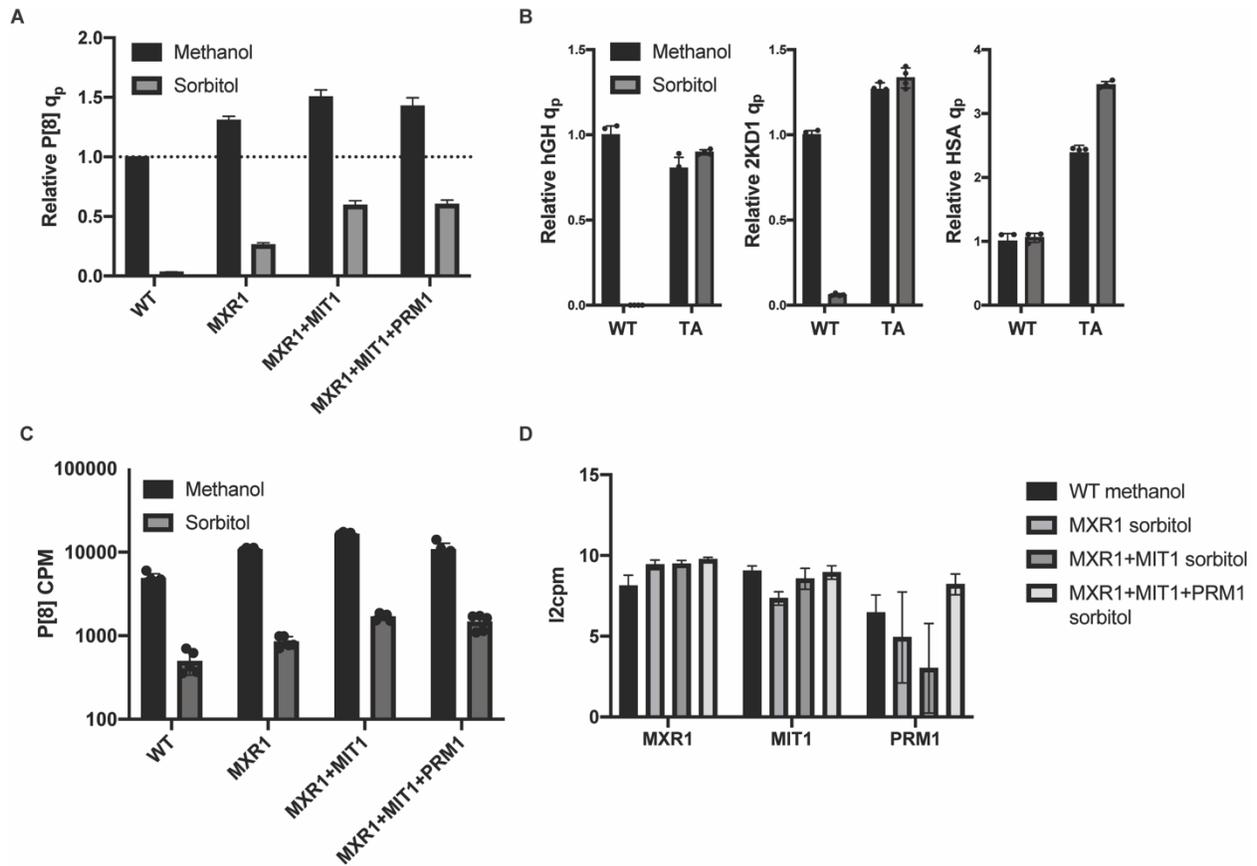
Figure 3.1 The P[8] productivity of strains expressing multiple pAOX1 activators may be limited by poor transgene expression

A) Comparison of the relative P[8] specific productivity of 4 strains fed with sorbitol or methanol in 3 mL batch cultivations. B) Comparison of the relative specific productivities of a WT and triple activator (TA, overexpressing MXR1+MIT1+PRM1) strain fed with sorbitol or methanol in 3 mL batch cultivations for 3 proteins of interest: human growth hormone (hGH), a nanobody (2KD1), and human serum albumin (HSA). C) Comparison of transgene expression levels for 4 strains expressing P[8] in methanol-fed or methanol-free 3 mL batch cultivations. D) Comparison of the methanol-free expression level of three known pAOX1 activators relative to a methanol-fed WT control.

Puzzled by the attenuated methanol-free production of P[8], we conducted RNA-seq analysis to better understand the source of this production challenge. We found that P[8] transgene activation in all sorbitol-fed conditions was lower than all methanol-fed conditions (**Figure 3.1C**). This result could not be attributed to insufficient expression of activating transcription factors, as expression of MXR1, MIT1, and PRM1 in the P[8] strain expressing all three TFs was greater than or equal to levels of expression observed in the methanol-fed WT control (**Figure 3.1D**). Thus, it appears that for the production of P[8], overexpression of all known transcriptional activators of pAOX1 is insufficient for full activation of pAOX1 in the absence of methanol. We wondered whether additional uncharacterized regulatory elements could be required to achieve comparable methanol-free production of P[8] in the absence of methanol.

3.2.2 RNA-seq guided CRISPR screen for methanol-free production of P[8]

In yeast, transcriptional regulation of enzymes involved in utilization of alternative carbon sources are often dependent on multiple TFs (Österlund, Bordel, & Nielsen, 2015; Soares, Antunes, Marinho, & Real, 2014; Soontorngun, 2017; Turcotte, Liang, Robert, & Soontorngun, 2010). These TFs can be regulated through post-translational modifications, like phosphorylation, or transcriptionally by altering levels of TF expression. This second regulatory mechanism can be readily assessed through RNA-seq and reconstituted through multiplex strain engineering to determine if specific TFs that are highly correlated with the methanol-utilization (MUT) pathway impact activation of pAOX1. We, therefore, began our search for uncharacterized regulatory elements by

examining how induction with methanol alters the transcription of TFs and methanol-utilization (MUT) pathway genes during the first 24 hours of protein production.

Two 24-well plates were inoculated with a WT strain expressing the model protein, P[8], and spaced 12 hours apart. 24 hours after inoculation, each plate was media exchanged into a production medium containing 5 v/v% methanol. Beginning 12 hours into production in the first plate, cell pellets were collected and frozen from both plates at 2-hour intervals. These samples were processed together, resulting in an RNA-seq dataset approximating the gene expression profile of a heterologous protein expressing WT culture over a 24-hour period of protein production. The resulting data showed wide variation in the expression of key MUT pathway genes, with methanol-induction occurring 12-14 hours into the production period (**Figure 3.2A**).

To identify potential regulatory candidates, we leveraged this dataset to identify strong potential interactions between TFs and the MUT pathway. We first calculated a gene-by-gene correlation matrix including all known TFs in Pichia logged in CIS-BP and MUT pathway genes defined by KEGG (Kanehisa et al., 2008; Weirauch et al., 2014). We then applied a Fischer transformation and filtered out weak interactions that fell between the 10th and 90th percentiles (two-tailed p-value > 0.2). This resulted in a sparse matrix consisting of 58 genes distributed across 3 groups obtained via hierarchical clustering (**Figure 3.2B**). Group 2 contained 7 highly correlated genes which were mainly related to methanol metabolism, including AOX1, MIT1, DAS1, DAS2, FLD1, and FGH1. We reasoned that additional regulators of pAOX1 would likely be highly correlated with genes in module 2, so we further refined our search by filtering out all genes that did not participate in a significant interaction with module 2, leaving a

set of 23 genes of interest. From this list, we selected 15 genes for overexpression or KO (**Figure 3.2C**).
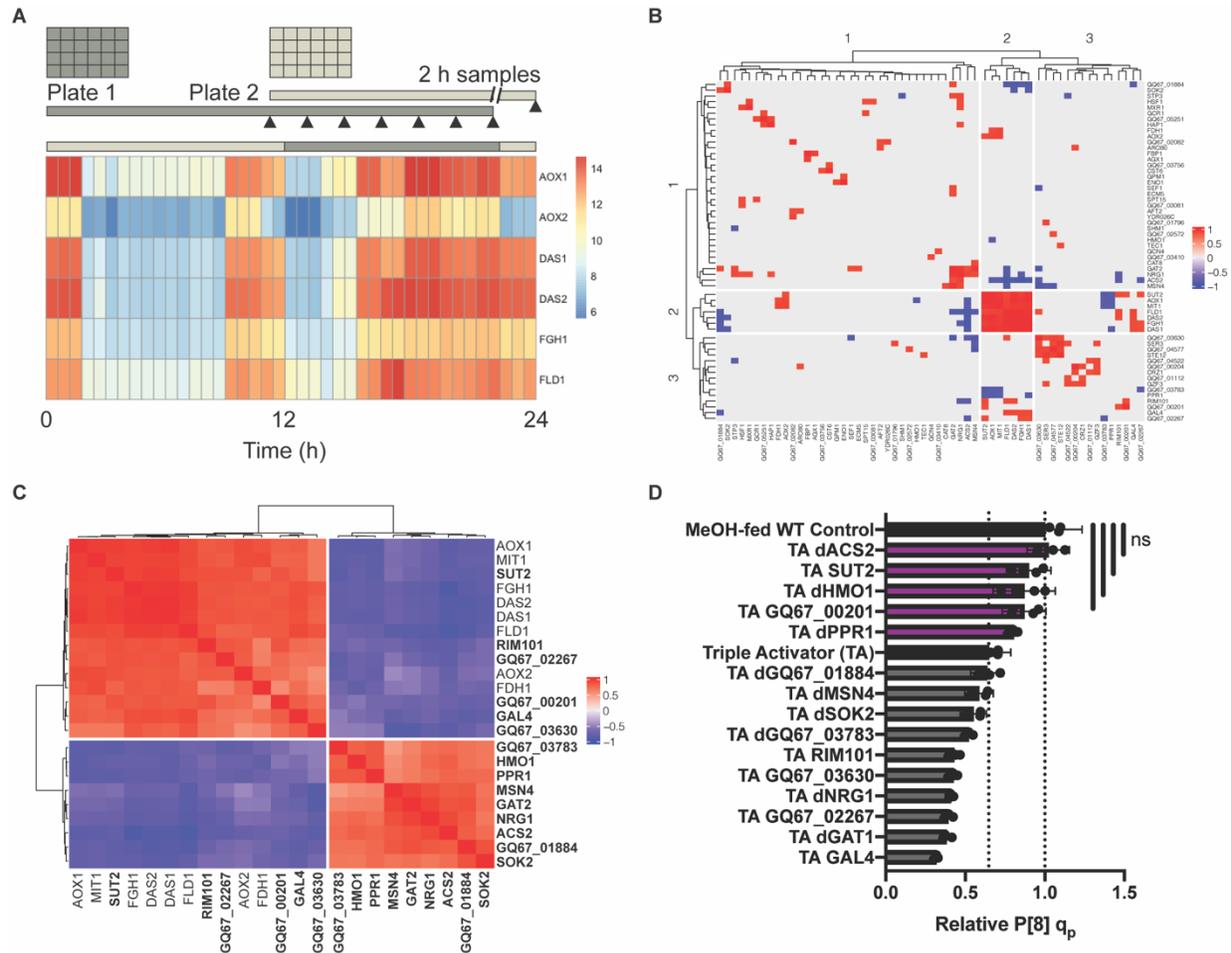
Figure 3.2 Transcriptomics aids the discovery of novel potential pAOX1 regulators

A) RNA-seq samples taken every 2 hours from 24-well plates spaced 12 hours apart enable reconstruction of MUT pathway induction over a 24 hour period, as evidenced by the transcriptomic profiles of key MUT pathway enzymes. B) Clustering of significant Pearson correlation coefficients (two-tailed p-value < 0.2) results in the identification of a highly correlated MUT pathway module, consisting of AOX1, MIT1, DAS1, DAS2, FLD1, FGH1, and SUT2. C) Pearson correlation coefficients of 23 genes found to be

significantly correlated with the MUT pathway module. D) Comparison of the relative P[8] specific productivity of strains containing one of 15 potential regulatory edits in a triple activator strain (TA, containing MXR1, MIT1, and PRM1) in sorbitol-fed 3 mL batch cultivations. 1-way ANOVA; ns: $p > 0.05$.

Beginning from the triple activator base strain, we upregulated or KO-ed genes of interest according to their positive or negative correlation with the group 2 MUT module and screened the impact of each genetic modification on P[8] expression in the absence of methanol (**Figure 3.2D**). Interestingly, 5 genetic modifications (dACS2, SUT2, dHMO1, GQ67_00201, and dPPR1) led to a statistically significant improvement (purple, adjusted p-values < 0.05) in the q_p of P[8] relative to the triple activator base strain and the q_p of the top 4 strains were not significantly lower than the methanol-fed WT control (adjusted p-values > 0.05). ACS2 and HMO1 impact histone acetylation and DNA packaging, suggesting a possible epigenetic component to pAOX1 activation. PPR1 and SUT2 are implicated in metabolic processes for pyrimidine and lipid metabolism, the latter of which has been shown to strongly impact protein productivity in Pichia during our previous media development studies (Biedermann, Gengaro, Rodriguez-Aponte, Love, & Love, 2021; Yang et al., 2020). GQ67_00201 has not been previously characterized. Due to its strong correlation with pAOX1 during induction on methanol, we elected to refer to it as Methanol Induced Transcription factor 2 (MIT2) hereafter.

3.2.3 Transcriptional characterization finds improved pAOX1 activation and P[8] transcription

Our initial transcriptomic screen yielded five genetic modifications that improve production of P[8] in the absence of methanol. We next used transcriptomics to assess the source of improvements in productivity. We first examined the impact of each mutation on transgene expression. We found that each mutation that increased the specific productivity of P[8] also resulted in higher transgene expression than the triple activator control strain (**Figure 3.3A**). Two mutations related to epigenetic regulation, dACS2 and dHMO1, had the strongest impact on transgene expression, resulting in methanol-free transgene expression levels exceeding the methanol-fed WT control. Surprisingly, however, only the strain overexpressing MIT2 also had significantly higher levels of AOX1 ($p < 0.05$) (**Figure 3.3B**), suggesting that the strong improvement associated with ACS2 or HMO1 KOs must be explained by an alternate mechanism. qPCR demonstrated that both ACS2 and HMO1 KOs resulted in significantly higher transgene copy numbers than the triple activator control ($p < 0.05$) (**Figure 3.3C**). Thus, the data suggest two distinct mechanisms to improve transgene expression: improved activation of pAOX1 by upregulation of MIT2 and improved transgene integration, aided most effectively by KO of ACS2 or HMO1.
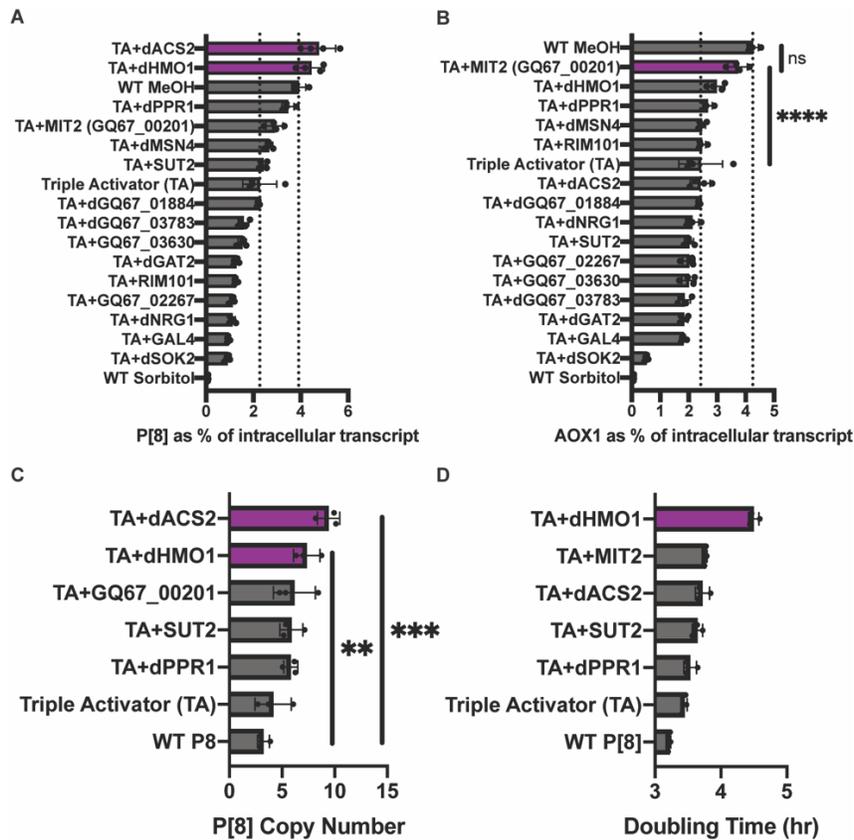
Figure 3.3 Transcriptomic and copy number analysis identify two potential mechanisms which may improve methanol-free expression of P[8]

A) Comparison of P[8] transgene expression in multiplex engineered strains under sorbitol-fed conditions relative to a methanol-fed WT control. B) Comparison of AOX1 gene expression in multiplex engineered strains under sorbitol-fed conditions relative to a methanol-fed WT control. C) Comparison of P[8] copy numbers for high performing strains demonstrates that KO of epigenetic regulators may significantly improve P[8] copy number. D) Comparison of growth rates for multiplex engineered strains in DM1 medium. 1-way ANOVAs; ns: $p > 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

The application of either mechanism suggested by our RNA-seq driven CRISPR screen was sufficient to improve transgene expression and drive comparable P[8]

productivity to a methanol-fed WT control. The resulting data suggest that multiplex methanol-free engineering may be applicable to many proteins of interest and transcriptomics may be used to further address any potential product specific challenges.

3.3 Discussion

Here we investigated the use of multiplex genetic engineering for optimal methanol-free production of heterologous proteins in *Komagataella phaffii*. In contrast to previous studies, which have largely focused on overexpression of one or two activating transcription factors at a time, we created and characterized the performance of a *K. phaffii* strain overexpressing all 3 known activating transcription factors: MXR1, MIT1, and PRM1 (Lin-Cereghino et al., 2006; Sahu et al., 2014; X. Wang et al., 2016). We found that this triple activator strain achieved methanol-free productivities comparable to a methanol-fed WT control for 3 of 4 proteins of interest, demonstrating that the benefits of methanol-free expression strategies in *K. phaffii* can be realized without attenuating specific productivity, a commonly cited issue in prior work (Chang et al., 2018; Vogl, Sturmberger, et al., 2018; J. Wang et al., 2017). Furthermore, guided by a novel transcriptomic timeseries dataset, we found that further genetic engineering could address methanol-free productivity challenges for a rotavirus derived subunit vaccine component (P[8]). While the difference in performance between P[8] production and the production of other heterologous proteins remains unclear, our methodology was able to resolve the methanol-free P[8] production challenge, resulting in 4/4 proteins with

comparable methanol-free production under methanol-free sorbitol fed cultivation conditions.

The COVID-19 pandemic has highlighted the need for improvements in biomanufacturing (Forman, Shah, Jeurissen, Jit, & Mossialos, 2021; Kim et al., 2021). *K. phaffii* has many advantages as a manufacturing host, but manufacturing capacity and bioprocessing conditions for *K phaffii* may be limited by its reliance on a methanol feedstock (Potvin et al., 2012). This work demonstrates that overexpression of multiple activating transcription factors is sufficient to achieve methanol-free productivity comparable to a methanol-fed wild-type control over a broad range of proteins. We expect such expression systems will continue to improve further as further advancements are made to multiplex engineered biomanufacturing hosts.

While this work provides support for the generalizability and benefits of using multiplex engineering to develop improved strains of *K. phaffii*, we also acknowledge certain limitations in the present study that may be addressed in future work. First, while it is clear that multiplex engineered strains of *K. phaffii* fed with a non-repressive carbon source can achieve specific productivities comparable to methanol-fed wildtype controls, it appears that the degree of engineering required may vary somewhat from protein to protein. Further understanding of the source of this variability would better inform strain and process design decisions. Secondly, the precise impact of each combination of transcription factors remains unclear, as the combinatorial complexity of multiplex strains currently precludes a more detailed understanding of the relative contribution of each component. Advances in cloning which facilitate easier multiplex expression of regulatory elements could be used to better understand the impact of

genetic engineering choices on the underlying host biology. Finally, while we identified

and conducted initial characterization of some novel beneficial mutations for P[8], more

detailed experimentation is necessary to understand the biological impact of the novel

genetic edits made in our CRIPSR screen. Such work may be the subject of future

studies.

Alternative hosts, such as *K. phaffii*, hold significant promise for improving the

production of protein biologics, with applications ranging from low-cost vaccines to

monoclonal antibodies (Brady & Love, 2021; Jiang et al., 2019; Matthews, Wright, et al.,

2017). The optimization of alternative hosts for large scale industrial bioprocessing,

however, may require some substantial engineering to introduce and optimize desired

phenotypes of interest, such as methanol-free production in the case of this study. This

work provides an early proof-of-concept that such multiplex engineering efforts can

achieve broadly applicable and industrially useful results. Further improvements to our

ability to predict and engineer beneficial multiplex mutations will enable further progress

in the design of next-generation manufacturing hosts.

3.4 Materials and Methods

3.4.1 Yeast Strains

All strains were derived from wild-type *Komagataella phaffii* (NRRL Y-11430).

Transcription factors MIT1, MXR1, and PRM1 were integrated into the genome near

genomic loci GQ67_02967, GQ67_04576, and GQ67_04382, respectively, using a

markerless CRISPR-Cas9 system described previously (Dalvie et al., 2019). All TFs

screened in the triple activator strain were integrated near the GQ67_03363 genomic

loci. All TFs were under the control of the pCAT1 promter from *K. phaffii*, and

sequences for pCAT1 and all TFs were amplified from the *K. phaffii* genome. Genetic

knockouts were conducted using via simplified genetic knockout using CRISPR-Cas9-

induced homologous recombination previously described (Dalvie, Lorgeree,

Biedermann, Love, & Love, 2021).

3.4.2 Cultivation conditions

All cultivations were conducted in 24-well deep well plates (25 °C, 600 rpm).

Cells were cultivated in Rich Defined Media, described previously (Matthews, Kuo, et

al., 2017a). An equal carbon mass of 40 g/L of sorbitol or methanol was used to test the

impact of carbon source selection on culture performance in RDM medium. A

comparison of growth rate was conducted in our recently improved medium for biomass

accumulation, DM1, described previously (Biedermann, Gengaro, Rodriguez-Aponte,

Love, & Love, 2022). All cultivations involving protein production were inoculated from

WCBs and outgrown to a target cell density of 20 OD600/mL, as described previously

(Biedermann et al., 2022). Supernatant samples were collected after 24 hours of

production, filtered, and analyzed.

3.4.3 Analytical methods

Supernatant titers for P[8] were measured by reverse phase liquid

chromatography as described previously (Biedermann et al., 2022), and normalized by

cell density as measured by OD600. Protein titer for other proteins of interest was

measured by size exclusion chromatography using an Agilent AdvanceBio SEC column

(300 Å, 2.7 μm) and 150 mM sodium phosphate running buffer fed at a rate of 0.25

mL/min. Copy number was analyzed using an NEB Luna universal qPCR Master Mix.

RNA extraction, sequencing, alignment and analysis was conducted as described

previously (Dalvie, Biedermann, et al., 2021a).

4. Modular development enables rapid design of media for alternative hosts

This work has previously appeared as: Biedermann, A. M., Gengaro, I. R., Rodriguez-Aponte,S.A., Love,K.R., &Love, J. C. (2022). Modular development enables rapid design of media for alternative hosts. *Biotechnol and Bioeng*. doi:10.1002/bit.27947

4.1 Motivation

Achieving high volumetric productivities of biologic drugs in cultivation is a key step in advancing candidate biologic drugs. The outcome of this effort ultimately impacts manufacturing costs as well as readiness for transitioning clinical-stage development (J. C. Love, Love, & Barone, 2012). The development of standard, chemically defined media for established manufacturing hosts, such as CHO, has made such transitions efficient for monoclonal antibodies by achieving high biomass accumulation, cell viability, operational consistency, and specific productivities, streamlining development efforts (McGillicuddy, Floris, Albrecht, & Bones, 2018; Rodrigues, Costa, Henriques, Azeredo, & Oliveira, 2012). Nonetheless, optimizing productivity or quality attributes for a specific product often still requires further refinement of media (Ritacco, Wu, & Khetan, 2018). Such development may require evaluating dozens of variants derived from a common standard formulation to address the specific challenges encountered, because the most sensitive media conditions are often not known *a priori* and testing many potential conditions at production-scales is costly and time-intensive (Gagnon et al., 2011; Loebrich et al., 2019). Media development for entirely new biomanufacturing technologies, such as alternative hosts

(Matthews, Kuo, Love, & Love, 2017b) or new product modalities (Lu et al., 2016), may also require new formulations or extensive optimizations due to limited prior knowledge.

Common approaches to develop a medium to optimize a phenotype of interest are often labor intensive, low throughput, or rely heavily on extensive analytical capacity (Galbraith, Bhatia, Liu, & Yoon, 2018). For example, analysis of residual media after cultivation requires extensive capabilities for analytical characterization and prior experience with the manufacturing host to identify potentially limiting or toxic media components (Mohmad-Saberi et al., 2013; Pereira, Kildegaard, & Andersen, 2018). As a result, optimizations can be slow and iterative. Furthermore, for an alternative host such as *Komagataella phaffii* (formerly known as *Pichia pastoris*), there is substantially less, if any, prior knowledge available to establish profiles for residual components in media after fermentation. Other analytical techniques like RNA-seq combined with methods for reporter metabolite analysis can guide media optimization, to generate testable hypotheses regarding beneficial modifications to media (Matthews, Kuo, et al., 2017b). Such genome-scale approaches, however, require prior host-specific knowledge, such as well-annotated genomes, and are still limited by slow iteration and labor-intensive preparations of new media to test the hypotheses generated from computational analyses. Additional methods use targeted measurements, such as qPCR quantification of marker genes (Burgard, Valli, Graf, Gasser, & Mattanovich, 2017) or elemental analysis (Carnicer et al., 2009; Tavasoli, Arjmand, Omid, Siadat, & Shojaosadati, 2017), to optimize media composition; although widely applicable, these methods do rely on secondary assay development and are generally not amenable to high-throughput experimentation to test wide ranges of conditions or components in a timely manner.

Alternative strategies for blending basal components for media allow linear combinations of existing media to explore many variations rapidly (Ardila-leal, Alvarado-ramírez, Guti, & Pedroza-rodríguez, 2020; Brühlmann et al., 2017; Jordan, Voisard, Berthoud, & Tercier, 2013; Vellanki, Potumarthi, & Mangamoori, 2009). While this approach avoids slow iterative analyses, the typical experiment is labor intensive to perform, often requiring independent preparations of over a dozen stock media to combine (Rouiller et al., 2013). Similar to analytical-based approaches for optimization, the selected variations of media are simultaneously guided and constrained by prior experience and media designs, which may limit the breadth of components examined (Kennedy & Krouse, 1999). For less established hosts with fewer available formulations of media, media blending may also require fully *de novo* formulations for initial studies. Further complicating such designs, different and new components for media can present challenges in solubility or unanticipated interactions with other elements in the formulations (Ritacco et al., 2018). New approaches to blending could, however, enable fast, flexible experimentation and minimize the time, labor, and analytical development needed initially to optimize media for new applications and phenotypes.

Here, we present a novel and generalizable approach for the modular development of media and demonstrate its use to create optimized media for two different phenotypes—cellular growth and recombinant expression of a protein (as measured by the secreted heterologous protein titer) from *Pichia pastoris*. Our approach comprises two modular parts for blending and optimization. We determined that a set of simple concentrated stock solutions constructed in defined modules could generate many media by blending or dilution. We then automated a simple, inexpensive liquid

handling system (Opentrons OT-2) to enable high-throughput screening for the effects of diverse media on a phenotype of interest in milliliter-scale batch cultures. To maximize the benefit of this automated blending, we also developed an algorithmic framework for systematic modular media optimization, beginning from a simple minimal media (here a YNB-based one). This framework provides insights pertaining to key media components during stages of optimization, as well as overall mapping of the design space for the media. In the examples presented here, the resulting defined media developed with this strategy outperformed commonly used BMGY and BMMY complex formulations for biomass accumulation and secreted heterologous protein production.

4.2 Materials and Methods

4.2.1 Strains and cultivation conditions

Media for evaluating biomass accumulation were developed using a previously described strain expressing G-CSF under control of the pAOX1 promoter (Crowell et al., 2020). 24-well plate screens were conducted as described previously, except cells were grown on a labForce shaker and were only sampled 24 hours after inoculation (Matthews, Kuo, et al., 2017b). BMGY, BMMY, and RDM media were formulated as described previously for shake flasks (Matthews, Kuo, et al., 2017b). All cultivations were inoculated from a working cell bank at an initial cell density of 0.1 OD/mL. For each working cell bank, cells grown in 1 L shake flasks with a 200 mL working volume of RDM were harvested during exponential growth (4-5.5 OD/mL) via centrifugation at 1500 rcf for 4 minutes at 23 °C and resuspended in an equal volume mixture of RDM

and 50 v/v% glycerol. This mixture was then distributed into 700 μL aliquots and stored at -80 °C, resulting in a cell density of ~30 OD/mL for the cell bank.

Media for evaluating enhanced production were developed using a strain expressing a rotavirus-derived subunit vaccine candidate, P[8], under the control of the pAOX1 promoter described previously (Dalvie, Brady, et al., 2020). Biomass accumulation proceeded for 24 hours; cells reached an initial induction density of ~20 OD/mL. Cultures were then exchanged into production media and allowed to produce protein for an additional 24 hours. Supernatant was harvested by centrifugation at 1500 rcf for 4 minutes at 23 °C and filtered using a Captiva 96 well 0.2 μm filter plate (Agilent Technologies, Santa Clara, CA) prior to titer measurement by RP-UHPLC.

Media components and supplements were purchased from Sigma-Aldrich, St. Louis, MO, unless otherwise indicated in our previously reported supporting information (Biedermann et al., 2022). A table of supplement and stock solutions with screening concentrations is also included in this supporting information. All components were suspended in aqueous solutions, although some commercial supplements did contain solubilizing agents or surfactants as detailed by their manufacturer. During modular optimization, all media were prepared in high throughput using an Opentrons OT-2 liquid handler (Opentrons, Brooklyn, NY, software version $\geq$3.16.1) using Openblend. Modular media blending code and instructions for setup and operation are provided in the Openblend package (https://github.com/abiedermann/openblend_public). For consistency, media used in final head-to-head comparisons were prepared in bulk and filter sterilized through a 0.2 μm benchtop filter.

4.2.2 Analytical procedures

Biomass was measured by optical density at 600 nm as described previously (Matthews, Kuo, et al., 2017b). An Agilent Bravo liquid handler was used to dilute samples prior to measurements of OD into the Tecan Infinite M200 Pro plate reader.

Reverse phase ultra-high performance liquid chromatography (UHPLC) analysis was performed on Agilent 1290 Infinity II UHPLC system controlled using OpenLab CDS software (Agilent Technologies, Santa Clara, CA). The concentration of protein was determined using a Poroshell 120 SB-Aq column (2.1 x 50 mm, 1.9µm) operated at 1.0 mL/min and 70 °C (Agilent Technologies, Santa Clara, CA). Buffer A was 0.1% (v/v) TFA in water and buffer B was 0.1% (v/v) TFA, 0.5% (v/v) water in ACN. A gradient was performed as follows: 30% B for 1 min., 30-40% B over 3 min., 40-90% B over 0.5 min., 90% B for 0.5 min., 90-30%B over 0.5 min., and 30% for 1 min.; total method run time was 6.5 minutes. Sample injection volumes were 50µL. A diode array detector was set for absorbance detection at 214nm. Data analysis was completed using OpenLab CDS Data Analysis (Agilent Technologies, Santa Clara, CA).

Statistical analysis and DOE design was conducted using JMP (SAS Institute, Cary NC). Quadratic models were fitted using effect screening and non-significant terms (adjusted p-value > 0.01) were eliminated sequentially in order of decreasing adjusted p-value to avoid overfitting. Data was plotted using Prism 8.4.0 (GraphPad Software, San Diego, CA). All replicates are biological replicates and a minimum of three replicates were used for each experiment.

4.3 Results

4.3.1 Design of approach for modular media blending

We sought to develop an approach capable of identifying important, beneficial

modifications for media tailored to a given phenotype of interest. We reasoned that key

requirements for such an approach would be that it is fast and automatable, with

minimal dependence on complex analytical assay development. Such features would

enable routine application to any measurable phenotype of interest. In general, media

blending allows both speed and low analytical complexity. We aimed to retain these

features while minimizing the labor and constraints on compositions imposed by linear

combinations of fully formed and unique media. We reasoned that diverse and flexible

blends of media could be created by defining simple concentrated stock solutions as

basic modules to combine further. These modules would comprise individual

components or common subsets of components with compatible solubilities (e.g. YNB).

If media components could be formulated in concentrated stock solutions that could be

stored stably over time, then the components could be routinely and interchangeably

combined and diluted to the desired final concentrations. This approach would yield a

broadly applicable modular strategy for media blending amenable to conventional liquid

handling automation.

To test the feasibility of this approach, we first assessed whether many common

media components could be formulated in concentrated stable aqueous stock solutions.

Using the CHO medium eRDF as a reference, we estimated the solubility of each

component of this medium, using data from AqSolDB as well as other online sources

(Combs, 2012; FSA Panel on Additives and Products or Substances used in Animal

Feed (FEEDAP), 2011; Ritacco et al., 2018; Schnellbaecher, Binder, Bellmaine, & Zimmer, 2019; Sorkun, Khetan, & Er, 2019; Yamamoto & Ishihara, n.d.). We compared the estimated solubility of each media component to its concentration in eRDF and found that, individually, most media components are soluble at levels >10x higher than their eRDF concentration (**Figure 4.1A**). The existence of a wide range of commercially available concentrated supplements further supports this result: >50x concentrated solutions of amino acids, vitamins, lipids, and trace metal supplements are common and commercially available.

Next, we used the product information of commercially available supplements, literature sources, and inspection to estimate the percentage of eRDF media components that could be stored in stable solutions for >6 months. We estimated that over 75% of eRDF components met this criterion (**Figure 4.1B**). To address stability challenges caused by less stable components, we reasoned that less stable components or supplements, such as vitamins, could be prepared, aliquoted, and stored frozen for long-term storage (Schnellbaecher et al., 2019); these aliquots could then be thawed and used within a defined period to mitigate component instability and enable their integration into our modular blending strategy. Together, these solubility and stability data suggested that a modular approach to media development could be defined in this way to accommodate a range of new formulations easily.

We next automated the process for constructing media, using the Opentrons OT-2. We chose this liquid handler due to its low cost, reliability, and compatibility with simple formats for data input, such as Excel spreadsheets. We then created an open-source Python package, named Openblend, which simplified the media construction

process by handling routine experimental design and execution steps (**Figure 4.1C**).

Openblend creates an experimental design spreadsheet, specifying the number of 24

well plates, the desired media composition of each well, and stock solution names and

concentrations. The script then checks the feasibility of the experimental design,

ensuring that the total volume of each well will not exceed the target volume and

avoiding the addition of sub-microliter stock solution volumes. If the design passes this

assessment, the script then outputs a new spreadsheet containing the setup for the OT-

2 deck and required volumes of stock solutions, providing a user with instructions on

how to setup the OT-2 liquid handler. We found that our typical time to execute this

script, setup the OT-2 and initiate plate building was ~15 minutes, and the time for the

automated steps was about two hours.

Figure 4.1 Modular media development can be broadly applicable, easily applied, and systematically executed to improve measurable phenotypes of interest

A) Estimate of the ratios of component solubility to their concentrations in medium demonstrates that most components are soluble at >10x their concentration in the CHO medium, eRDF. B) With the exception of some classes of medium components, such as vitamins, most media components can be formulated into solutions that remain stable for >6 months under proper storage conditions. C) Overview of time, labor and planning saved by using Openblend to automate modular media construction. D) Overview of a

modular media optimization approach, which can be used to build an optimized medium for any measurable phenotype of interest systematically.

Finally, we defined a modular approach for optimization to effectively leverage the Openblend tool (**Figure 4.1D**). Beginning from an initial basal medium, improved media are constructed through successive rounds of optimization. In each round, a library of media components and supplements are screened to identify beneficial additives. These additives are then screened in combination and over a range of concentrations to further optimize the performance of the medium. Each modular addition and optimization of additives can be guided simply by measurements of the phenotype of interest (e.g. biomass accumulation). This greedy approach to multi-dimensional optimization could continue iteratively until the resulting media met desired specifications, all available media components were explored, or no additional gains in performance realized.

4.3.2 Application to developing a medium for biomass accumulation

To assess the utility of this blending-based approach, we next aimed to identify and optimize the concentration of media components beneficial for rapid biomass accumulation of *P. pastoris* in batch cultivation. We previously described a rich defined medium (RDM) (Matthews, Kuo, et al., 2017b), capable of high growth rates during biomass accumulation. One challenge encountered with this formulation, however, was that precipitates can form at higher pH values that require filtering during bulk preparations (W. Zhang, Sinha, & Meagher, 2006). Nonetheless, this medium provided

a relevant comparison for assessing the medium realized with our new approach due to its prior demonstrated benefits relative to complex media. Following our modular approach, we improved biomass accumulation by optimizing the accumulated optical density at 600 nm after 24 hours of cultivation.

Algorithms for optimizing systems based on multiple dimensions are often sensitive to initial conditions used (Zakharova & Minashina, 2015). Given this potential confounding effect here, we tested first the effects of the types of carbon source, nitrogen source, and pH set point on biomass accumulation, using 1x YNB without amino acids or ammonium sulfate (YNB) to satisfy minimum requirements for the concentrations of trace elements. We conducted a full-factorial DOE using glycerol, glucose, and fructose as carbon sources; urea and ammonium sulfate as nitrogen sources; and potassium phosphate as a buffer with pH values of 5, 5.75, and 6.5. We selected initial concentrations of 40 g/L, 4 g/L urea or the N-mol equivalent for ammonium sulfate, and 10 g/L potassium phosphate, similar to values used in other media for *Pichia pastoris* (Matthews, Kuo, et al., 2017b). A least squares regression model, including individual, combination, and quadratic effects was fit to the log of optical density after 24 hours, a proxy variable for the average growth rate ($R^2 = 0.81$). We determined that the two most significant model terms were the type of carbon source and the interaction of the nitrogen source with pH (**Figure 4.2A**). We found that cells grew significantly faster on metabolically related sugars (glucose and fructose) than on the polyol (glycerol) commonly used for Pichia during biomass accumulation (**Figure 4.2B**). This result affirms prior reports where glucose has been used for biomass accumulation of Pichia (Guo et al., 2012; Moser et al., 2017).

The model also suggested that poor biomass accumulation during cultivation resulted from a combination of ammonium sulfate as a source of nitrogen with low buffer pH (**Figure 4.2C**). This outcome may result from the production of acidic species associated with cellular ammonium metabolism in the batch cultivation (Villadsen, 2015). Interestingly, the model indicated slightly greater biomass was achieved with urea instead of ammonium sulfate, in agreement with prior literature (Guo et al., 2012). The biomass accumulation of cultures grown with urea as a source of nitrogen were less sensitive to reduced pH values (~5). We observed, however, that cultivations at pH 5 showed extensive flocculation compared to those at 6.5. Given the insensitivity of urea-fed cultivations to buffer pH and the high solubility and potential for low-cost sourcing of fructose, we therefore chose to include fructose, urea, and a potassium phosphate buffer with a pH of 6.5 in our initial media formulation.
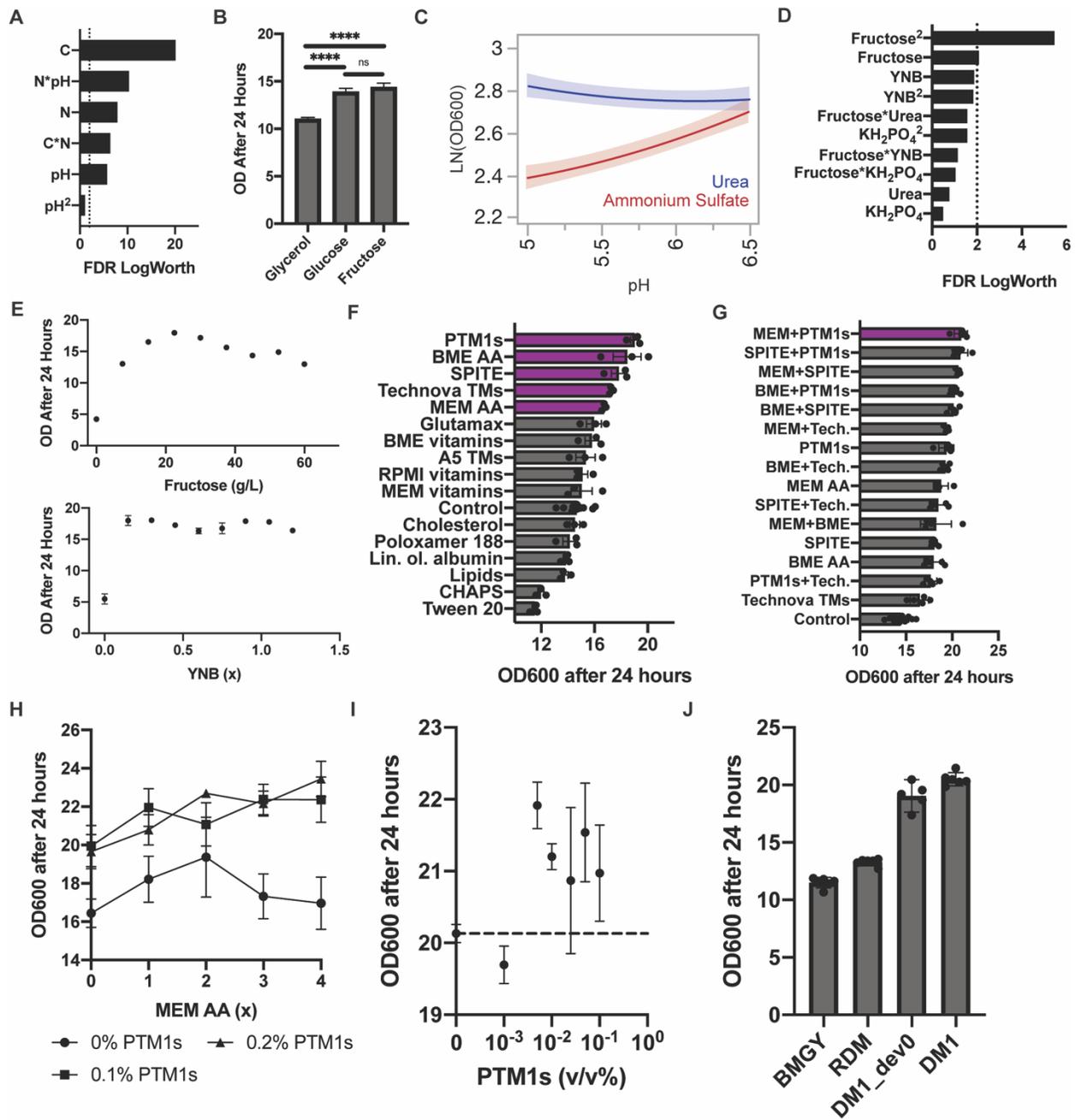
Figure 4.2 Modular development of a new biomass accumulation media for *K. phaffii*
A) Significance of carbon (fructose, glucose, glycerol), nitrogen (urea and ammonium
sulfate), and pH choice (5, 5.75, 6.5) in a least square regression model fitted to a full
factorial DOE. B) Fructose and glucose were found to result in significantly higher
biomass accumulation after 24 hours of outgrowth than glycerol. C) Ammonium sulfate
was found to be more pH sensitive than urea, as shown by the JMP sensitivity profiles

during fructose feeding. D) Significance of terms in a least square regression model fitted to a full factorial DOE over fructose, urea, potassium phosphate, and YNB concentrations. E) 1-FAAT optimization of fructose and YNB concentration finds optimal outgrowth performance at a fructose concentration of 22.5 g/L and relative insensitivity over a wide range of YNB concentrations (0.15 to 1.2x). F) A media supplementation screen identified 5 beneficial supplements, related to trace element and amino acid supplementation. G) Further screening of beneficial supplement combinations identified synergistic amino acid and trace metal supplementation strategies. H) Comparison of the effect of MEM amino acid concentration on biomass accumulation at different PTM1 salts concentrations. I) Effect of the concentration of PTM1 salts on biomass accumulation in DM1_dev0 medium supplemented with 1x MEM AA. J) Head-to-head comparison of 4 v/v% glycerol BMGY, 4 v/v% glycerol rich defined medium, the initial defined biomass accumulation media (DM1_dev0), and the final biomass accumulation medium obtained after a full optimization cycle (DM1), demonstrates that DM1 leads to superior biomass accumulation.

With this basal formulation determined, we next screened for concentration-dependent interactions of other key additives to the media and then optimized concentration-dependent parameters. Following the same approach for screening effects, we conducted a full factorial DOE over a broad range of media component concentrations: YNB (0.5, 1, 2x), fructose (10, 30, 50 g/L), urea (1, 4, 7 g/L), and potassium phosphate adjusted to a pH of 6.5 (4, 10, 16 g/L). The resulting model identified fructose as a concentration-sensitive parameter (R^2=0.73) (**Figure 4.2D**).

Terms involving the concentration of YNB were also highly ranked, but not statistically significant. No significant interactions between components were identified in the model. We therefore sought to better understand the concentration dependence of fructose and YNB independently (**Figure 4.2E**), over an 8-fold range of concentrations. As expected, biomass accumulation was highly sensitive to fructose concentration, with an optimum around 22.5 g/L of fructose. The concentration of YNB had minimal effect on biomass accumulation; the presence of trace elements supplied by YNB, however, was essential to growth. Based on these results, we chose concentrations of 22.5 g/L fructose, 1x YNB, 7 g/L urea, and 10 g/L potassium phosphate buffer. We reasoned that although biomass accumulation was relatively insensitive to the concentrations of YNB and urea, higher concentrations could provide improved media depth in future applications. We named this basal formulation DM1_dev0.

We next assessed what additional media components could improve biomass accumulation. To test over 60 different components individually would require over 60 individual solutions. Such an approach would scale linearly with new components; instead, we chose to screen groups of related components, using commercially available pre-mixed supplements. We compiled a library of 16 commercial supplements and industrially-relevant surfactants containing more than 60 unique components and screened their individual effect on biomass accumulation after 24 hours. In this way, we reasoned we could efficiently identify critical classes of components related to the phenotype of interest and potentially deconvolve specific individual additives of interest by inference. We used the recommended concentrations of each supplement as supplied in product information, or critical micelle concentrations, and prior knowledge

for broad classes in yeast media to set reasonable screening concentrations (Biedermann et al., 2022). We identified five beneficial and two detrimental supplements that significantly impacted biomass accumulation ($p_{adj} < 0.02$; 1-way-ANOVA) (**Figure 4.2F**). In general, the results suggest that supplementation with amino acids and trace metals were beneficial for accumulating biomass, while two surfactants, Tween 20 and CHAPS, were detrimental. For this phenotype, the effects of vitamin and lipid supplements were minor; supplements from either supplement category were not significantly beneficial or detrimental to biomass accumulation. Our earlier experiments suggest that vitamins are essential but concentration agnostic (**Figure 4.2E**), while lipid supplementation provides no clear benefit for biomass accumulation.

Based on these results, we chose to test whether combinations of supplements of amino acids and trace salts could yield synergistic improvements in biomass accumulation. We screened pairwise combinations of the five beneficial supplements of mixed composition and ranked the performance of our supplementation strategies (**Figure 4.2G**). A combination of 1x MEM amino acids with 0.1 v/v% PTM1 salts resulted in the highest yield of biomass, though we observed strong performance from other combinations of amino acid and trace metal supplements. Based on these data, we chose to add MEM amino acids and PTM1 salts in our basal medium and optimized their concentrations (**Figure 4.2H**).

Based on these results, we elected 0.1 v/v% PTM1 salts and 1x MEM amino acids, in order to balance the moderate benefits and potentially high costs of amino acids. We found, however, that the inclusion of the PTM1 salts in liter-scale preparations produced fine precipitates, which can impede sterile transfers in use. To

overcome this challenge, we screened a broad range of PTM1 salts concentrations to

identify the minimum concentration required for improved outgrowth performance

(**Figure 4.2I**). We found that PTM1 addition at concentrations as low as 0.0005 v/v% led

to increased biomass accumulation. We therefore revised our PTM1 salts concentration

to 0.01 v/v%, a concentration high enough to obtain the benefits of PTM1

supplementation without inducing precipitate formation. This formulation we named

DM1.

Completing this series of optimizations with our iterative modular approach to

define a new formulation of medium, we then compared with other common media used

to grow *P. pastoris*. We evaluated the performance of this new optimized medium

(DM1) relative to the unsupplemented basal medium (DM1_dev0), the rich defined

medium (RDM) we had previously developed, and a common medium 4 v/v% glycerol

BMGY. We found that DM1 yielded the highest biomass accumulation, with significantly

higher biomass accumulation relative to RDM and BMGY (**Figure 4.2J**). This result

demonstrates the utility of our modular strategy here for media development that yielded

an improved formulation for biomass accumulation compared to other common media

with minimal time and labor investment, and without requiring complex analytical

methods like mass spectrometry or RNA-sequencing.

4.3.3 Identifying media conditions important to heterologous protein production in *K. phaffii*

In addition to the time and labor savings of modular media development, our

proof-of-concept experiments demonstrated that this approach creates a flexible

medium that can be rapidly adapted to new growth phenotypes, as well as a data

package that identifies media conditions important to the phenotype of interest. We reasoned that these additional benefits could be particularly relevant for optimizing production of heterologous proteins. Understanding which media components contribute most significantly to productivity could improve culture performance and help identify important metabolic pathways or physiological effects for further study.

To develop a medium for improved production of a recombinant protein, we chose to use a strain engineered to secrete a rotavirus-derived subunit vaccine component, VP4-P[8], as a model protein. We have previously demonstrated that this viral antigen can be expressed at high titer under the control of the methanol-inducible pAOX1 promoter in BMMY media (Dalvie, Brady, et al., 2020). Similar to our initial approach to optimize a medium for growing biomass, we first determined and optimized the concentrations of the sources for carbon and nitrogen, along with the pH. The expression of P[8] in the strain tested uses the methanol-dependent pAOX1 promoter for inducible expression, so we selected methanol as the initial carbon source. We then examined the impact of the source of nitrogen and buffer pH on titer. We conducted a full-factorial DOE using identical concentrations as those used to create a medium for accumulating biomass. The resulting model was visualized by ranking combinations of sources of nitrogen and buffer (**Figure 4.3A**). The effects showed no interaction between these two factors. Urea was again identified as the preferred source of nitrogen while higher pH values led to improved secreted P[8] productivity. Unlike biomass accumulation, this pH dependence was observed across both nitrogen sources.

We next applied the same DOE to identify important concentration-dependent interactions that impact the production of P[8]. Unsurprisingly, the concentration of methanol was the most important factor, with possible minor effects from other components (**Figure 4.3B**). We decided to screen further a 20-fold range in methanol concentrations using two formulations for remaining media components—the one determined for optimal cell growth (DM1) and the optimal base media formulation predicted by the quadratic model here (2x YNB, 1 g/L urea, 4 g/L potassium phosphate adjusted to a pH of 6.5). We found that production was relatively insensitive for concentrations of methanol ranging from 1-4 v/v%, with an optimum around 2% (**Figure 4.3C**). This result is in reasonable agreement with previous Invitrogen guidance recommending methanol concentrations below 2 v/v% (Invitrogen Corporation, 2002). We postulated that the rapid decline in productivity observed in these milliliter-scale cultures using concentrations >6 v/v% methanol was likely due to excess formation of toxic metabolic byproducts such as formaldehyde and hydrogen peroxide (Wakayama et al., 2016). Interestingly, the predicted optimal medium from this set of studies outperformed the medium we determined for accumulating biomass, suggesting that certain components of the basal medium may benefit protein expression more than cellular growth and underscores the value of optimizing media for specific phenotypes of interest. Based on these data in total, we defined a basal medium for production including 2x YNB, 2 v/v% methanol, 1 g/L urea, and 4 g/L potassium phosphate buffer adjusted to a pH of 6.5 (DM2_dev0).

Next, we examined which supplements could improve the performance of DM2_dev0. We added three chemical chaperones (TUDCA, sodium deoxycholate

monohydrate (SDM), and valproic acid) (Kuryatov, Mukherjee, & Lindstrom, 2013; Uppala, Gani, & Ramaiah, 2017), two antioxidants (reduced glutathione (GSH) and N-acetyl cysteine (NAC)), and the chelator, K-ETDA, to the list of 16 supplements included in our original screen defined for biomass accumulation. Concentrations for these components were chosen based on product specifications, literature data, and prior experience. Many of the 22 supplements screened improved production of P[8] (**Figure 4.3D**). The top four ranking supplements comprised surfactants or lipids, which could modulate membrane fluidity and lipid metabolism (Butler, Huzel, Barnab, Gray, & Bajno, 1999; Degreif, Cucu, Budin, Thiel, & Bertl, 2019; Ritacco, Frank V; Yongqi Wu, 2018).

We then screened combinations of lipid supplements and surfactants to identify potential synergistic effects. We ranked the individual supplements and their combinations (**Figure 4.3E**) according to the measured titers of P[8]. We found that the addition of a cholesterol-rich supplement yielded the highest secreted titers of P[8] (~50% improvement compared with supplement-free condition in initial screens). Interestingly, a synthetic cholesterol supplement alone did not substantially improve performance, suggesting the benefit results from a combination of fatty acids and surfactant components in the supplement (**Figure A.2**). This conclusion is consistent with similar improvements observed from other supplements, such as linoleic acid-oleic acid-albumin (**Figure 4.3D**).

Since no other synergistic effects were observed in the combination screen, we assessed the dependence of titer on the concentration of the cholesterol-containing supplement identified (**Figure 4.3F**). Similar to our observations with cellular YNB used in the outgrowth media, we found that concentrations of the supplement as low as 0.2

v/v% were beneficial for protein expression, but that production was relatively insensitive to concentration (**Figures 4.3F, 4.3G**). We then directly compared the supplemented medium to the original composition; the new supplemented media provided a 25% improvement in titer (p = 0.0006, one-tailed Welch's T test). This new formulation with 1x cholesterol supplement, which we named DM2_dev1, was the result of one cycle of optimization using our method.
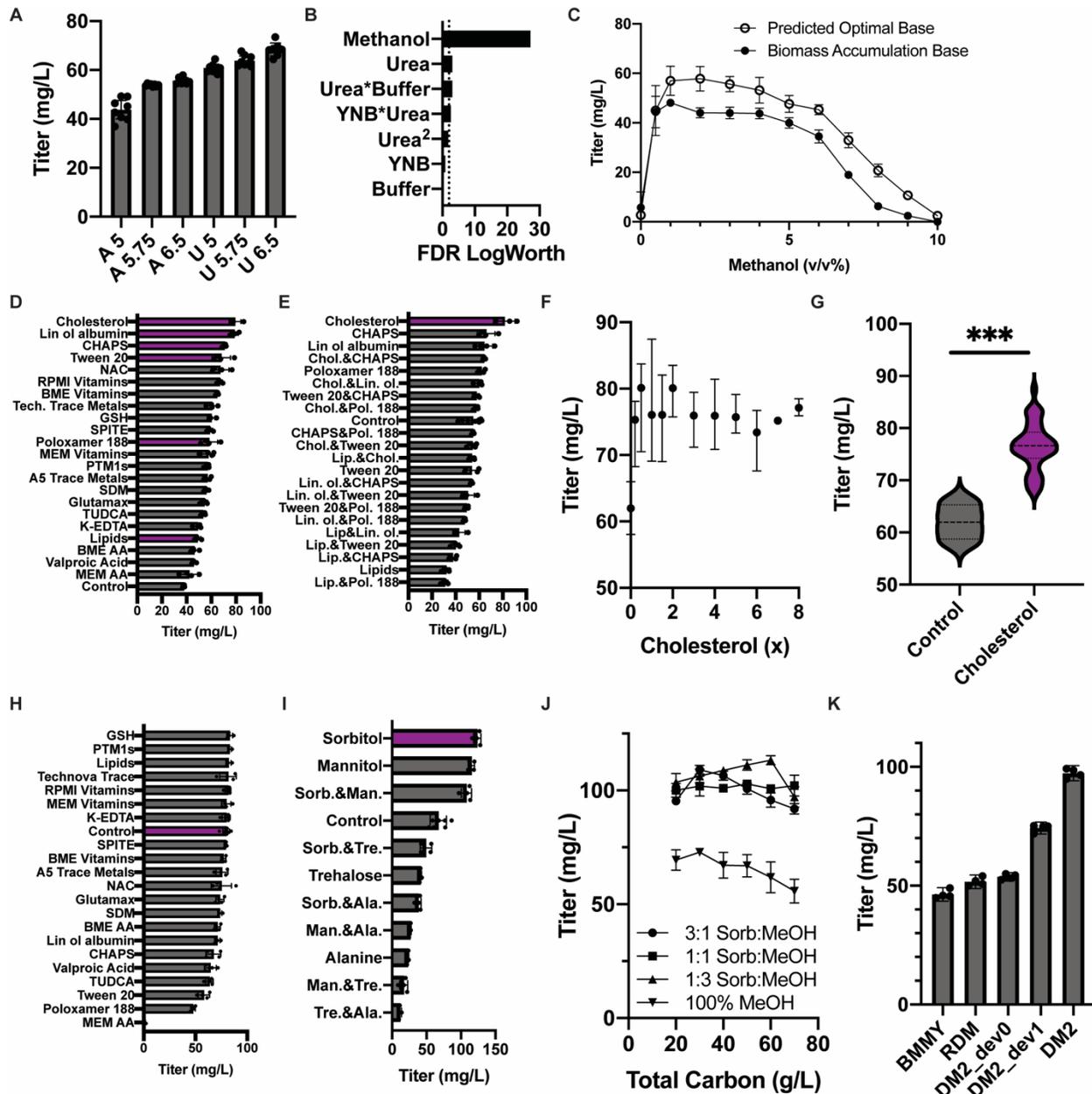
Figure 4.3 Modular development of a media for heterologous protein production in *K. phaffii*

A) Initial full-factorial screen of nitrogen source choice and buffer pH demonstrates that urea is preferred over ammonium sulfates and high buffer pH is preferred over lower values. B) A full-factorial concentration optimization identified methanol as the most concentration dependent variable. Other components in the base media were predicted to affect productivity with much lower levels of significance. C) Evaluation of the effect of

methanol concentration on P[8] titer, using two different base media (urea, buffer, and YNB concentrations): the biomass accumulation base medium and the optimal base media composition predicted by our concentration DOE. D) Ranking of supplements according to their effect on P[8] titer. Supplements related to membrane fluidity or lipid metabolism ranked highly. E) Evaluation of combinations of lipid and surfactant supplements confirmed that cholesterol supplementation leads to the greatest improvement in P[8] titer. F) Concentration optimization of cholesterol demonstrated low concentration dependence, with similar performance observed over a 40-fold range (0.2-8x). G) Comparing cholesterol-free and cholesterol-supplemented cultures fed at various concentrations demonstrates that cholesterol supplementation results in a significant ~25% improvement in P[8] titers (p<0.001). H) No significantly beneficial supplements were observed when repeating the supplementation screen. I) Screening supplementation of 20 g/L of co-fed substrates individually or in 1:1 combinations by mass identified sorbitol supplementation as highly beneficial to P[8] titer. J) Examination of the effect of co-feed ratio and total carbon concentration on titer in DM2_dev1 supplemented media. K) Comparison of P[8] titer obtained with DM2 to previous iterations and other common *P. pastoris* media demonstrates a ~2x improvement in P[8] titer, relative to 1 v/v% methanol RDM and 1 v/v% methanol BMMY.

Components of the cholesterol supplement included fatty acids, cholesterol, and cyclodextrin, which are all are known to modulate membrane fluidity, a key parameter in vesicle trafficking (Cooper, 1978; Degreif et al., 2019; Mahammad & Parmryd, 2015). We reasoned that the addition of this supplement could therefore have synergistic effects with other supplements, but did not find any further supplementation that improved P[8] titers within our original screen (**Figure 4.3H**). We, therefore, considered if there could be additional classes of beneficial supplements, absent from the original screen. Previous experiments demonstrated that P[8] productivity is highly sensitive to methanol concentration (**Figure 4.3C**), so we wondered whether further modulation of central carbon metabolism could yield additional productivity gains.

Modification of central carbon metabolism is best accomplished by feeding cells alternative carbon sources, either entirely or as co-feeding substrates. Four co-fed substrates have previously been shown to be non-repressive of pAOX1: sorbitol, mannitol, trehalose, and alanine (Inan & Meagher, 2001). These substrates can be co-utilized with methanol without repressing the pAOX1 promoter, which controls expression of P[8]. We hypothesized that the introduction of supplemental carbon sources could enable further optimization of central carbon metabolism. We screened co-fed substrates individually and in 1:1 combinations at a total concentration of 20 g/L (a concentration similar to the optimal fructose and methanol concentrations observed in previous carbon source optimizations) (**Figures 4.2E,4.3C**). Sorbitol co-feeding had the most beneficial effect, resulting in a ~80% increase in P[8] titer (**Figure 4.3I**). Mannitol supplementation was also beneficial (~70% increase), while alanine and trehalose co-feeding were detrimental to productivity. While co-feeding carbon sources

led to increased biomass yield during production, these differences did not account for the improved titer, as improvements in specific productivity (q_p) of ~60% and ~45% were also observed for the sorbitol and mannitol co-fed conditions, respectively (**Figure A.3**). Based on these data, we chose to include sorbitol as a supplemental carbon source for further study.

The addition of a supplemental carbon source could significantly impact central carbon metabolism. We, therefore, wondered how the inclusion of sorbitol might impact the optimal carbon feeding strategy. Examining total carbon source concentrations from 20 – 70 g/L, we compared the performance of cultures co-fed with sorbitol:methanol ratios of 3:1, 1:1, and 1:3 to a methanol-only control (**Figure 4.3J**). All co-fed conditions outperformed the methanol-only control, suggesting that the presence of sorbitol is highly beneficial for producing P[8]. The titer was relatively insensitive to sorbitol:methanol ratios and carbon concentrations. Based on the data, we decided to use 2 v/v% methanol and 20 g/L of sorbitol for the final sorbitol-supplemented media named DM2.

Finally, we compared the P[8] titer obtained using DM2_dev0, DM2_dev1, and DM2 to other common production media for *P. pastoris*: BMMY and RDM. We found that DM2 led to a ~2x improvement in P[8] titers, relative to BMMY and RDM, up to 97 $\pm$ 2 mg/L.

4.4 Discussion

Here we have implemented a novel and broadly applicable approach for media development that relies on rapid, automated construction of diverse media from defined

modules of components. We demonstrated the utility of this approach by developing two

new media for two phenotypes of interest in the heterologous production of proteins by

yeast, namely biomass accumulation and secreted production. We systematically

identified and optimized the concentration of media components important to each

phenotype of interest. Importantly, defining these new formulations of media did not

require advanced analytical capabilities and required minimal experimental time to

assess more than 360 total formulations during two to three rounds of optimization for

each.

Our optimized formulations affirmed the importance of lipid-related components

for maximizing titers in *Pichia pastoris* cultivations. The importance of optimizing

membrane fluidity or lipid metabolism has been well established in CHO and appears to

be key to optimizing heterologous protein secretion in *P. pastoris* cultivation as well

(Clincke et al., n.d.; Ritacco et al., 2018; H. Zhang, Wang, & Liu, 2013). This result adds

further support to prior work which identified the importance of ergosterol, antifoam, and

Tween 20 or 80 for production of other heterologous proteins in *P. pastoris* (Apte-

Deshpande, Rewanwar, Kotwal, Raiker, & Padmanabhan, 2009; Baumann, Adelantado,

Lang, Mattanovich, & Ferrer, 2011; Routledge, Hewitt, Bora, & Bill, 2011). This work

also broadens knowledge of media development for *P. pastoris* by providing a map of

sensitive media parameters, furthering contributions by studies of other media

formulations which have been previously reported for biomass accumulation or for

testing the influence of one or a related set of components (Farinha, Araújo, & Freitas,

2019; Ghosalkar, Sahai, & Srivastava, 2008; Kaushik, Lamminmäki, Khanna, & Batra,

2020; Marsalek, Puxbaum, Buchetics, Mattanovich, & Gasser, 2019; Totaro et al., 2021).

Modular media blending has four advantages over existing methods. First, the use of common stock solutions and supplements to formulate media reduces initial labor required for new experiments or optimizations ~15 minutes per experiment, making parallel testing of multiple hypotheses efficient and requires less resources overall. Here, we created 30 stock solutions, and evaluated >360 unique media compositions, without manual preparation of individual media or extensive blending calculations or planning. Most of these solutions could be readily reused in future experiments to optimize for new phenotypes of interest. Second, our method requires minimal knowledge of the host organism *a priori* and could, in principle, be applied to any measurable phenotype of interest. We anticipate that this method could be used to optimize other phenotypes of interest, such as glycosylation profiles. Third, our method provides certain practical advantages, including minimal requirements for analytical characterization and rapid identification of component interactions that lead to solubility challenges. These traits make it possible to learn about formulations that may lead to extensive precipitates like those encountered with our rich defined medium formulation (**Figure 4.4A**). Finally, modularly constructed media, such as DM2, can be ~70% pure water with low osmolarity, leaving volumetric and osmotic space for future modifications to accommodate new or related phenotypes of interest (**Figure 4.4B**).
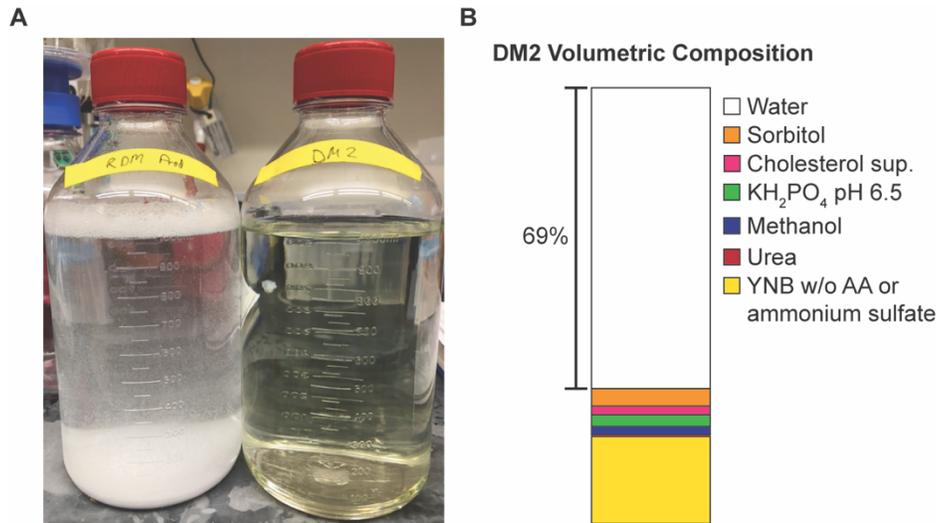
Figure 4.4 Comparison of DM2 to rich define medium

A) Comparison of precipitate formation during construction of RDM (left) and DM2 (right) media. Adjusting the pH of RDM to 6.5 results in significant formation of white precipitate. No precipitate formation is observed in DM2. B) Relative volumes of stock solutions and pure water needed to construct DM2. Pure water addition accounts for 69% of DM2 volume, demonstrating that there is substantial room for further supplement exploration and development. When separated into simple stock solutions, DM2 can be 3x concentrated.

We also acknowledge certain limitations in the present study that may be addressed in future work. First, while modular media development identifies components key to the optimization of the phenotype of interest, additional media optimization effort may be necessary to translate these learnings in batch cultivations to scaled-up fed-batch or perfusion operation, where additional variables such as supplemental feed composition and feeding schedule must also be considered. In principle, modular media construction could be applied to high-throughput scale-down

cultivation models, such as Ambr250s, and learnings from plate-scale experiments could be used to guide further evaluations of candidate media for different operating modes in bioreactors like intensified fed-batch or perfusion based on the most influential components identified. Second, our approach for optimization relies on greedy algorithms tailored to create a new media for a single phenotype of interest; however, given the vast explorable media space it is possible to find a local optimum. Further metabolic or -omic modeling techniques could be employed to guide broader exploration of media space, co-optimize multiple phenotypes, or facilitate biologically informed optimization, albeit with more complex experimental and computational requirements (Matthews, Kuo, et al., 2017a; Mohmad-Saberi et al., 2013). Third, our current method used commercially available supplements, but in practice, beneficial supplements could be simplified by using individual components, to facilitate more biological inferences and aid development of improved host-specific supplements. Finally, initial screens to identify beneficial supplements rely on reasonable choices of initial concentrations for screening. These currently require prior knowledge from the literature or commercial sources; with further use in the community of the Openblend approach, it is possible additional sharing of knowledge could help inform further developments. We also anticipate that such knowledge sharing may also clarify the generalizability of DM2 for optimal production of any protein of interest and establish rapid approaches to adapt DM2 for optimal production of new products of interest.

The improved speed and accessibility of in-depth media development experiments enabled by modular media construction could help improve expression of many classes of proteins in laboratories and discovery centers that have not traditionally

77

had access to such capabilities. An automated platform provided in the Openblend package may aid such efforts—although the principle of modular construction from concentrated stock solutions could still be leveraged in the absence of automation equipment. Since many lead candidates for new therapeutic proteins begin in small biotech firms and academic labs, early-stage improvements in productivity could help advance more proteins towards the clinic simply by facilitating access to larger quantities of proteins for initial research and non-clinical studies. In more established companies, the ability to make rapid improvements to existing media may enable faster product development timelines and could reduce manufacturing costs overall. Rapid identification and optimization of sensitive media components could also enable easier adoption of a range of industrially relevant alternative hosts, resulting in further manufacturing flexibility and potentially cost savings (Coleman, 2020). The rapid development of new blends may also help facilitate a deeper understanding of the metabolic and physiological impact of media components on host biology when used in combination with additional analytical methods.

5. Initial progress on the computational design of an ultra-low cost protein nanoparticle vaccine platform for immunogenic display of viral antigens

This work is based on ongoing unpublished research conducted in collaboration with Shuting Shi, who has aided pT3 design scouting, algorithm development, and new pT3 model redesign efforts.

5.1 Motivation and project conceptualization

The choice of *K. phaffii* as a production host introduces trade-offs in subunit vaccine design. *K. phaffii* can grow up to 8x faster than a mammalian expression host, enabling rapid development and highly productive, scalable manufacturing (Matthews, Wright, et al., 2017); however, as a lower eukaryote, *K. phaffii* is generally more limited than mammalian expression systems in the size and complexity of proteins it can produce at high titers, with smaller simpler proteins produced at higher titers than larger proteins with complex tertiary or quaternary structures—although much work is still ongoing to improve alternative host performance (Jiang et al., 2019). Thus, currently, *K. phaffii* can be used to rapidly produce the 20 kDa SARS-CoV-2 receptor binding domain (RBD) but would likely struggle to produce a significant quantity of 450 kDa trimeric spike protein used in the Novavax NVX-CoV2373 vaccine, to use subunit components derived from SARS-CoV-2 as an example. Smaller subunit components, such as RBD, tend to have poorer immunogenicity than large subunit components and often require

multivalent display on viral-like particles (VLPs) to induce the strong B-cell activation and high titers of neutralizing antibodies that lead to protective immunity (Sompayrac, 2012). Most available VLP designs, such as the commonly used HepB antigen, however, are aggregation prone and cannot be secreted at high titers, resulting in challenges in upstream (USP) and downstream (DSP) bioprocessing (Bale et al., 2015; Moleirinho et al., 2020). As a result, VLP production can be a rate-limiting step in bioprocessing, requiring low-yield intracellular expression that fails to realize many of the advantages of using *K. phaffii* as an expression system, driving up costs and slowing vaccine development and production rates. Upon closer examination, however, we believe that VLP manufacturing could be greatly improved through the design and integrated development of a two-component VLP optimized for secreted production in a co-culture of *K. phaffii* strains.

Two-component nanoparticles could enable significant improvements in VLP manufacturing if they are optimized for secreted production by *K. phaffii*. Multicomponent nanoparticles are common to many viral capsid structures and symmetry groups (Montiel-Garcia et al., 2021; Twarock & Luque, 2019). Recently, researchers have shown that it is even possible to computationally design novel two-component VLPs by symmetrically docking naturally occurring multimers based around icosahedral symmetry groups (Bale et al., 2016; Cannon et al., 2020; King et al., 2014; Wargacki et al., 2021) (**Figure 5.1A**). Some of these computationally designed nanoparticles can even be assembled in vitro from their constitutive components (**Figure 5.1B**) (Wargacki et al., 2021). In light of this prior work and the natural ability of *K. phaffii* to secrete small proteins at high titer, we expect that it should be possible to

achieve strong secreted production of a VLP with *K. phaffii* because this problem can be simplified to two separate but coupled protein design and secretion problems by using a *K. phaffii* co-culture; given two protein components of reasonable size that are predicted to form a nanoparticle, we can computationally design a practically infinite number of protein variants and screen these variants for strong expression by *K. phaffii* and for extracellular assembly into a VLP during a *K. phaffii* co-culture cultivation. Thus, by integrating experimental and computational workflows, we should be able to design an effective VLP platform optimized for production by our chosen host, *K. phaffii*.

Secreted production of a VLP simplifies and accelerates vaccine development and manufacturing in several ways. First, secreted VLP production could reduce processing time, cost, and complexity by increasing titer and eliminating process steps such as cell harvest, centrifugation, lysis, and refolding and simplifying purification. Second, secreted production could enable compatibility with intensified bioprocessing methods such as perfusion, further boosting volumetric productivity and decreasing costs. Third, and most intriguingly, secreted production of a VLP could potentially enable secreted production of a multi-component subunit vaccine in a single one-pot co-culture (**Figure 5.1C**), containing strains expressing each of the two VLP components and a strain expressing a pathogen-derived antigen, such as RBD, enabling the extracellular formation of 400+ kDa VLP complexes. *K. phaffii* produces few host cell proteins, so these 400+ kDa VLP complexes could be expressed in over 80% pure supernatant. It seems possible that such a production method could leverage the significant size difference between the product and impurities, enabling purification by

nanofiltration or simple size exclusion chromatography which would yield large cost savings and further simplify downstream processing.

Having conceived this promising project direction, in this chapter we present initial progress towards the realization of this novel VLP production method. We describe initial proof-of-concept studies and discuss improvements in nanoparticle design strategy and implementation. As this is also a new research direction for the lab, key technical hurdles passed are also detailed in greater depth in the **Appendix**.
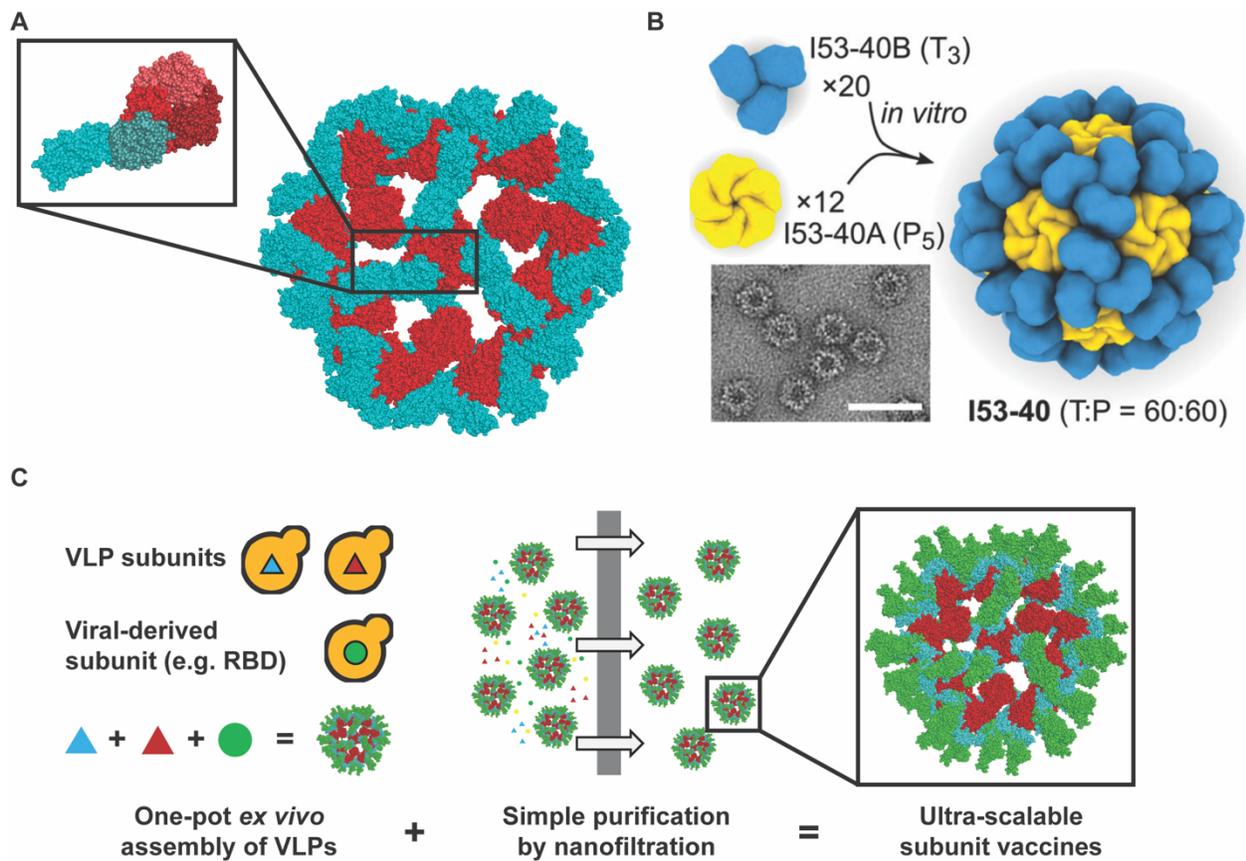
Figure 5.1 *K. phaffii* co-cultures could enable secreted production of 2-component VLP-based vaccines

A) Visualization of the IPD-designed I32-28 nanoparticle, constructed from a dimer and trimer placed at 2- and 3-fold icosahedral symmetry axes (adapted from Bale et al., 2016). B) Schematic illustrating that de novo designed particles can be assembled in

vitro from their constitutive components (sourced from Wargacki et al., 2021). C)

Overview of a potential one-pot method for secreted production of VLP-based vaccines

using a *K. phaffii* co-culture.

5.2 Results

5.2.1 Initial re-design work provides key proof-of-concept

With our goal of achieving secreted production of a 2-component protein

nanoparticle in mind, we first wondered whether there could be previously designed 2-

component nanoparticles in which both components could be efficiently secreted. We

began by screening several existing nanoparticle designs from the literature (Bale et al.,

2016; King et al., 2014); however, as shown by **Figure 5.2A**, only 1 protein out of 20

tested achieved even moderate secreted titers. The poor initial expression of previously

reported nanoparticles suggested that further efforts would likely be necessary to tailor

nanoparticle designs for optimal expression by *K. phaffii*.

As a proof-of-concept for this approach, we wanted to test if we could improve

the expression of an IPD-designed nanoparticle component using Rosetta, a widely

used protein design software which uses a Monte Carlo algorithm based on a heuristic

scoring method to optimize protein structures (Leman et al., 2020). Our lab had not

previously used this design tool, so we were also interested in understanding the

technical feasibility and pipelines necessary to scale such protein design efforts. We

reasoned that a better understanding of our scout culture success rate and the

experimental workload and cycle time would help us understand if a larger, focused

VLP design effort could be successful.

We hypothesized that improving hydrophilicity by reducing hydrophobic content and increasing protein charge could enable improved titers. Excessive hydrophobicity can reduce titers by increasing the likelihood of non-specific intracellular aggregation. Non-specific aggregation drives the formation of Lewis bodies that are then passed to the proteosome for degradation, reducing the fraction of intracellular product which successfully transits the secretory pathway (Amani & Naeem, 2013; Sankar, Krystek, Carl, Day, & Maier, 2018). Rosetta uses amino acid reference energies to control amino acid composition during design simulations (Jacak, Leaver-fay, & Kuhlman, 2011). This appears to result in average hydrophobic contents of ~55%, based on previous VLP designs released by IPD. In contrast, proteins such as human serum albumin (HSA) that can be produced at very high titers tend to have higher hydrophilicity, ~40% hydrophobic amino acids in the case of HSA. In addition, increasing protein charge shifts the pI of proteins away from the near neutral pH experienced during intracellular protein folding, leading to surface charge that could further increase hydrophilicity (Der et al., 2013).

Starting from the I32-28 nanoparticle structure, we redesigned the individual components to improve the hydrophilicity by adding hydrophobic content and charge biasing terms to a symmetric Rosetta design simulation. As shown in **Table 5.1**, the individual components of the 2-component nanoparticle achieved significant reductions in hydrophobic content, as measured by GRAVY, % of hydrophobic amino acids, and the magnitude of net charge. We did not achieve these improvements at the expense of component or particle stability, as total energies associated with these structures was lower in redesigned models than in the original structure. The resulting change in

surface characteristics are shown in **Figure 5.2B**. By running successive simulations,

we were also able to achieve a significant diversity in protein sequence. As shown in

Figures 5.2C&D, each design generated had less than 65% sequence identity from

simulations where we disallowed redesign of intercomponent interface residues

(percent identity among designs was as low as 20% when interface residue design was

allowed). We found this result exciting because a wide range of accessible sequence

space could de-risk production challenges associated with any sequence-specific

issues.

We next generated 15 nanoparticle designs, including 8 constraining the

interface residues and 7 allowing redesign of any residue. As shown in **Figure 5.2E**,

several of the redesigned subunit resulted in significantly improved productivity, with 3-4

of the 15 designs screened demonstrating strong secreted production rates. Strong

expression of subunit B was not observed, although this could have been due to the

strong +20 positive charge constraint or challenges with intracellular aggregation or

dimerization.

Overall, these results provided an initial proof-of-concept by demonstrating that a

potential VLP component could be redesigned to enable high secreted expression by *K.*

phaffii. Further improvements to design target metrics, strategy, and workflow, however,

appear necessary to improve the likelihood of success and facilitate rapid screening of

novel nanoparticle designs.

Design	Total Score	A GRAVY	B GRAVY	A Instability Index	B Instability Index	A Score	B Score	A Net Charge	B Net Charge	A % Hphobic	B % Hphobic
I32_original	-60491	0.368	0.282	41.1	33.6	-434	-422	-8	-7	56.1	56.0
I32_0.5_0.1	-132456	-0.158	-0.177	48.1	10.6	-818	-710	-17	18	45.9	42.0
I32_0.5_10	-141575	-0.173	-0.259	42.9	14.9	-443	-333	-19	17	43.9	36.6
I32_1.0_0.5	-105280	-0.111	-0.530	54.1	27.2	-768	-488	-19	18	45.9	35.3

Table 5.1 Redesign algorithm improves nanoparticle stability and reduces hydrophobicity
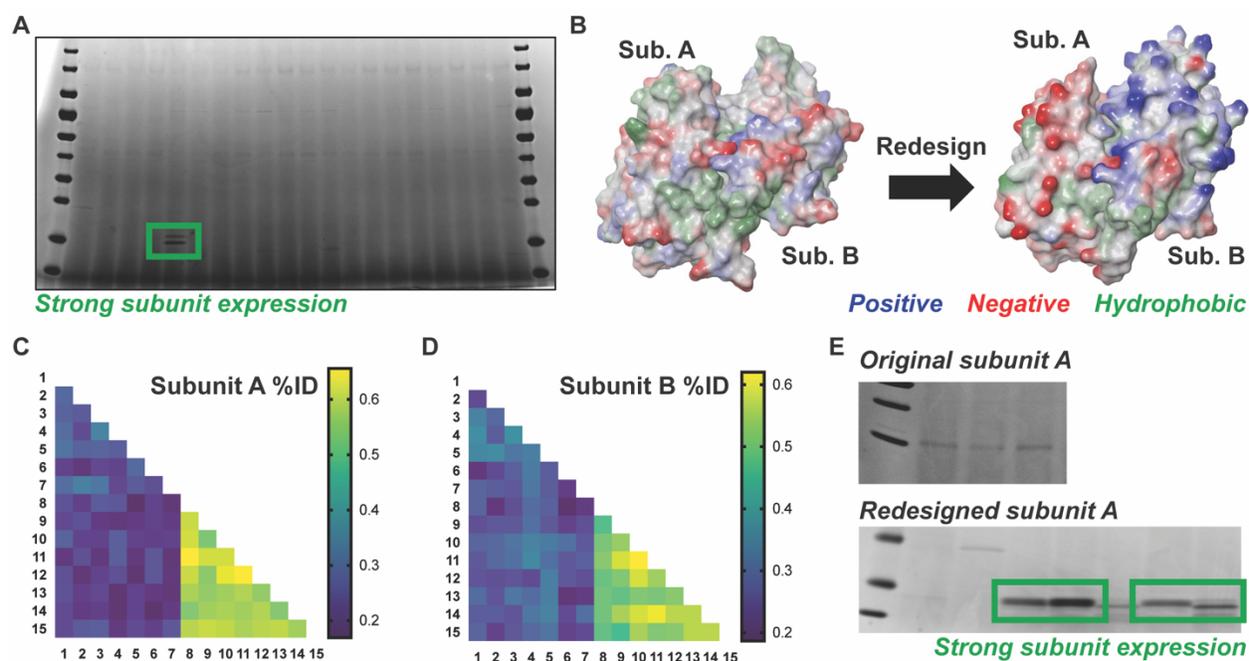
Figure 5.2 Redesigned I32-28 nanoparticle demonstrates proof-of-concept improvement in secreted production of a nanoparticle subunit

A) Protein expression as measured by SDS-PAGE obtained from an initial scout culture of 20 IPD nanoparticle subunits derived from 10 VLP designs. B) Visualization of surface hydrophobicity and charge before and after redesign with Rosetta. C) Pairwise percent identify of subunit A amino acid sequences from the set of I32-28 nanoparticle redesigns. D) Pairwise percent identify of subunit B amino acid sequences from the set of I32-28 nanoparticle redesigns. E) Protein expression observed in initial scout cultures of I32-28 subunit A before and after redesign as measured by SDS-PAGE.

5.2.2 Identifying challenges for the development of a secreted protein nanoparticle

Following initial proof-of-concept experiments, we next considered how preliminary work could be expanded to increase the likelihood of successfully expressing both components of a 2-component nanoparticle and to accelerate design and screening workflows. We first considered risks and challenges that were encountered during our proof-of-concept experiment.

We realized that the construction of hetero-multimeric VLPs from smaller homo-multimers (as done by previous IPD work with dimers, trimers, and pentamers) may complicate expression and present screening challenges. First, homo-multimers introduce quaternary structure which may increase protein folding complexity and reduce the likelihood that a component can be secreted at high titer. Second, successful nanoparticle construction requires that each component first forms a homo-multimer, introducing an additional screening step beyond the initial scout cultivation. Finally, homo-multimers make it more challenging to distinguish between product-related variants and host cell proteins in an SDS-PAGE gel, necessitating further follow-up experiments. For example, in the case of a trimer, partial reduction in SDS-PAGE could lead to three different expected molecular weights, corresponding to a monomer, dimer, and trimer. When one then considers that each subunit in the homomultimer could contain sequence modifications, such as incomplete signal peptide cleavages or glycosylation, the range of potential molecular weights for a product-related variant becomes quite broad, increasing the likelihood that further experiments beyond the initial scout cultivation will be needed to determine if a product candidate should be slightly altered or completely redesigned when a band of unexpected molecular weight

is present. Thus, alternatives to current VLP designs which contain homo-multimers should be explored.

In addition to protein design considerations, we observed limitations in our initial computational workflow. First, nanoparticle redesign simulations required extensive computational capacity; a single simulation contained 120 proteins and required 1-2 days of runtime on compute resources with 8-16 GB of RAM. We purchased access to commercially available high-performance computing (HPC) resources; however, our usage was limited by system administrators to ensure equitable sharing of available computational resources. While this is a completely reasonable policy for an HPC provider, it could significantly delay design cycle times, as a round of simulations that could take two days would require 1-2 weeks. Second, down selecting nanoparticle designs requires scalable calculation of relevant metrics. Initial redesign efforts required manual evaluation of several potentially important metrics, such as AggScore, which introduces an additional bottleneck in the protein design process that extends cycle times and limits the number of designs that can be considered for experimental validation.

We also observed limitations in our initial experimental workflow. On the experimental end, this project requires routine screening of batches of 30-40 proteins, each requiring the construction, cloning, and linearization of 30-40 plasmids. While this can be done with a BioXp which automates Gibson assemblies, such a process is potentially error prone and requires manual DNA clean-up and linearization. In practice, this can introduce delays due to BioXp run failures and poor yields after DNA recovery or linearization. The reliance on BioXp plasmid preparation may also necessitate follow-

up Sanger sequencing experiments to confirm the fidelity of transformed plasmids, as it may not otherwise be clear whether poor expression reflects challenges secreting a protein or a cloning failure. The large number of proteins screened in a single batch of protein sequences increases the likelihood of encountering any of these issues, further increasing costs and extending timelines.

5.2.3 Addressing challenges for the development of a secreted protein nanoparticle

A novel starting structure with pT3 symmetry may improve titers and enable faster design screening

As discussed in the previous section, we wanted to determine whether large protein nanoparticles could be designed without reliance on initial homo-multimerization interactions. Initially, this seemed like a significant challenge, as most VLPs are constructed from homo-multimers rotated about icosahedral symmetry groups; however, we noticed that some viruses contain a unique pseudo-T3 (pT3) symmetry that could be harnessed to enable VLP design without homo-multimeric interactions.

As shown in Figure **5.3A&B**, viral capsids structures can be classified according to their symmetry groups, which reflect the unit cell spacing between pentamers in the icosahedral structure. T3 viral capsids feature proteins arrayed across a truncated icosahedron and are constructed from homo-pentamers and homo-hexamers. pT3 capsids (**Figure 5.3C**) are deemed pseudo-symmetric because the homo-hexamer in a T3 capsid is replaced by a hetero-hexamer. This hetero-multimeric rotation that appears in pT3 symmetric capsids may endow viral capsids derived from this class with unique design and production advantages, as other pT169, pT21, pT25, pT27, and pT31

pseudo-symmetries yield considerably larger and more complex capsid structures that would be difficult to redesign using existing computational tools (Montiel-Garcia et al., 2021).

By removing the homo-pentamer and introducing targeted truncations to existing viral capsids (**Figure 5.3D**), we realized that it could be possible to design 25-30 nm diameter VLPs, assembled from 14-25 kDa monomers (which *K. phaffii* can produce at very high secreted titers), achieving our goal of eliminating homo-multimers from VLP design. We further reasoned that the ability to redesign existing monomers could facilitate substantial improvements in secreted titer and enable routine optimization of nanoparticle design to mitigate or evade anti-VLP immune responses. To our knowledge, pT3 symmetries have not previously been designed with Rosetta, so we developed custom computational tools to enable the creation of a corresponding symmetry file. The documentation on constructing complex symmetry files de novo is somewhat sparse, so a more detailed discussion of this process is included in the **Appendix**. The resulting files enable computational redesign of any pT3 viral capsid.
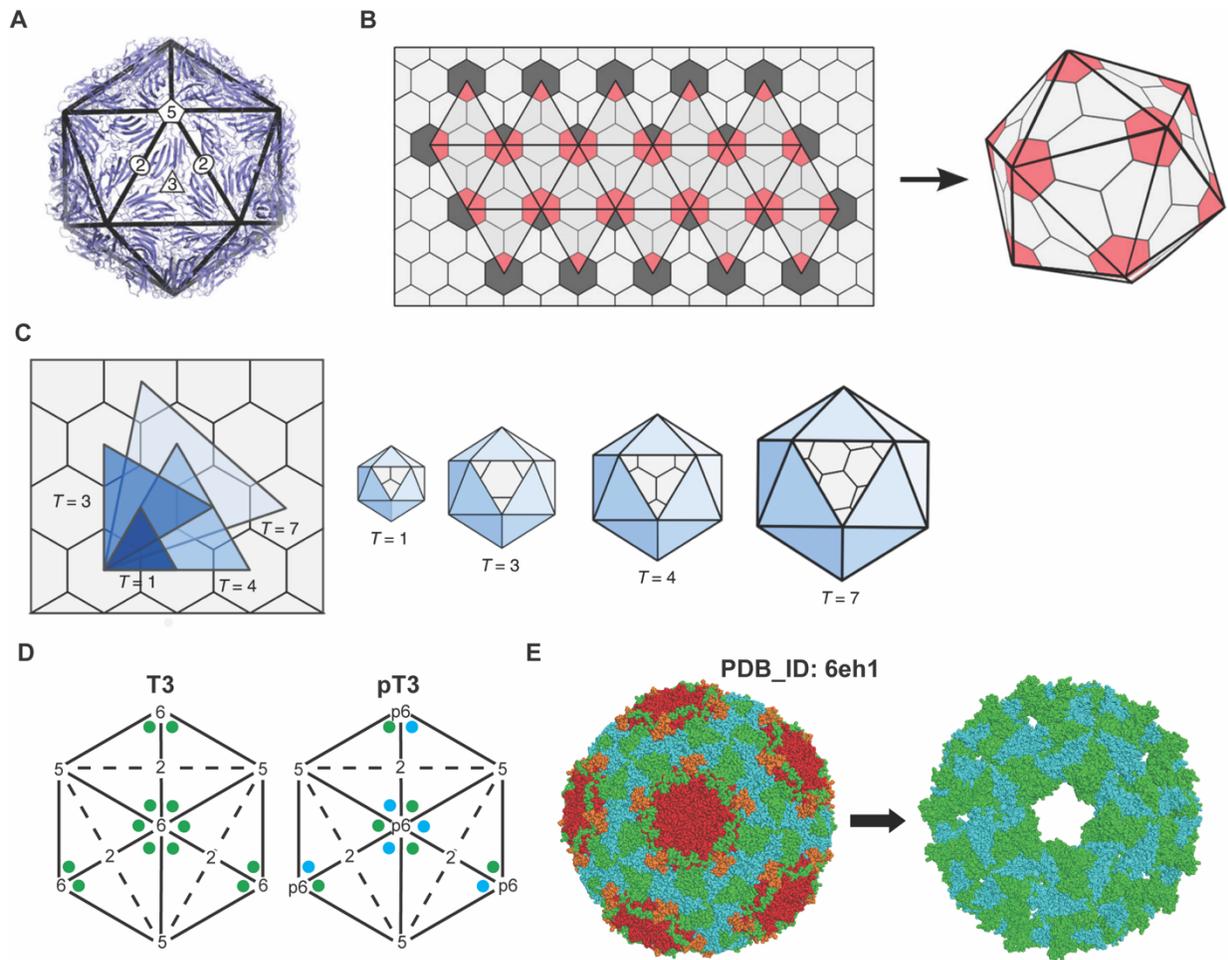
Figure 5.3 A unique viral capsid symmetry enables the design of VLPs without homo-interactions

(adapted from Figure 1 of Twarock & Luque, 2019)

A) Illustration of a viral capsid structure in which protein subunits are placed about the symmetric axes of an icosahedron. B) Schematic illustrating that icosahedral symmetry can be derived from a hexagonal lattice. C) Demonstration that different unit cell definitions result in different orders of icosahedral symmetry, denoted by T-numbers. D) Illustration that pT3 symmetry differs from T3 symmetry by replacing a homo-hexamer with a hetero-hexamer. E) Schematic demonstrating that components of a pT3 viral

capsid, 6eh1, can be removed, resulting in a nanoparticle constructed primarily by heterogenous interactions between components.

A rapid computational pipeline for scalable design generation and screening

As noted previously, our reliance on commercially available HPC resources significantly bottlenecked early computational efforts. Given the recent advent of cloud-based computing resources, we attempted to resolve these issues by implementing a scalable slurm scheduling system on Amazon Web Services (AWS). We basically wanted to create a computational environment similar to an HPC cluster, without any wait times or capacity limitations for computational resources. Such a cluster would have a head node where jobs could be submitted to our own custom job queue, and the queue's cluster of compute nodes would scale dynamically to recruit the necessary resources for each job.

There are many available computing products provided by AWS and it took some time to learn how to work effectively within their cloud computing environment, as it differs from traditional HPC clusters in a few key ways. First, the home directory of AWS is a transient memory storage location, so files left in the home directory are automatically deleted when a compute node is shut down. We solved this issue by creating an FSX file system and linking it to each new EC2 instance as they were created, either by specifying this link in the parallel cluster config file or by entering the required commands in the "User data" data field during step 3 of a manual instance launch (see **Appendix** for further details). Second, AWS has several similar systems to enable large scale parallelization of computation, so it was somewhat challenging to

identify the best approach for queue and cluster creation. We ultimately found that the AWS ParallelCluster CLI enables the creation of an intuitive and scalable parallel computing environment that is compatible with known HPC schedulers, such as slurm. We did, however, find that specific flags, such as "--exclusive" needed to be added to ensure proper allocation of sufficient RAM to each computational job by AWS (see **Appendix** for further details). Finally, special attention had to be paid to security group permissions to avoid communication issues between users and AWS applications (see **Appendix** for further details).

Improvements to our computational setup enabled the rapid creation of dozens of novel pT3 nanoparticles with high protein sequence variability (i.e. each design was only ~50% identical to any other design). We next had to develop a more scalable method for down selecting nanoparticle models with scalable analytical tools. Although initial pT3 models had the potential to be interaction free, we found that the design process could introduce or remove existing interactions. We also found that visual inspection of intermolecular interactions was slow, non-scalable, and error prone, as Pymol visualization of interactions was not always indicative of the energetic favorability of the interaction. We therefore implemented scripts to automate interface analysis in Rosetta, allowing us to classify nanoparticle designs by the presence and strength of desired or undesired interactions between subunits. Next, we implemented scripts which calculate dozens of relevant protein metrics using Rosetta or Schrodinger. These scripts yield detailed monomer specific characterization, providing information on aggregation propensity via Schrodinger's AggScore metric, as well as other relevant field such as Rosetta energy, isoelectric point, molecular weight, disorder propensity,

and hydrophobic composition. Properly installing the Schrodinger license, protecting the license server compute node from accidental termination, and configuring security groups correctly was a particularly challenging aspect of this analysis pipeline, so further discussion of these technical details is included in the **Appendix**. While experimental validation and further improvements in theoretical understanding will no doubt improve down selection tools, this initial set provides a strong basis for rapid down selection of promising design candidates. Analysis results generated with these tools are demonstrated further in the proceeding sections.

Accelerating experimental screening workflows with commercial partners

Improvements to our computational workflow enabled the rapid design and down selection of dozens of promising nanoparticle candidate designs. This led to an experimental bottleneck, due to the challenges of scaling error-prone in-house plasmid cloning and linearization. These problems would only be compounded by an increased design throughput, so we looked for a commercial partner which could deliver high quality linearized plasmids for our designed proteins of interest. We ultimately found that Genscript could provide this service at rates competitive with our previous BioXp workflow. Given a fasta file of protein sequences, Genscript consistently provides linearized pPICZαB vector containing each protein of interest, codon optimized for expression in *K. phaffii* and ready for transformation within 3-4 weeks. This allows experimenters to stay focused on protein designs and evaluating experimental validation, avoiding low value troubleshooting of intermediate steps.

5.2.4 Initial redesign and screening of pT3 nanoparticles

Multimerization and a constraint issue led to poor initial results

In parallel with the computational and experimental improvements implemented above, we began initial redesign efforts on a pT3 nanoparticle. In principle, we could have chosen any pT3 nanoparticle, but we decided initially to redesign the B and C components of the Sacbrood viral capsid (PDBID 6eh1). Without many of the tools we subsequently developed, we reasoned that the symmetry alone might reduce homo-multimerization propensity enough to enable successful expression of each component and potentially enable assembly of capsid structure *ex vivo*.

As shown in **Figures 5.4A&B**, we generated 13 new designs, each with less than 40% sequence identity to any other design. Subsequent DNA cloning with our BioXp workflow and expression characterization, however, did not result in successful expression of a protein derived from either the B or C components (**Figures 5.4C&D**). Puzzled by these poor early results, which conflicted with our earlier proof-of-concept work with I32-28, we searched for experimental and computational issues that might explain the initial lack of protein expression.

We first conducted an analysis of intracellular protein expression and used an anti-histag antibody and automated western blot Jess machine to assess whether our product of interest was present intracellularly. In some samples for component C, we found that the product was present intracellularly, but at higher molecular weights than would be expected for monomer expression (**Figure 5.4E**). This result suggested that multimerization may be occurring intracellularly, driven by the small interactions still present between C components.

We also compared the experimental structures generated by our initial pT3 redesign algorithm to the original experiment structure. In contrast to our previous proof-of-concept experimental results, we found that our initial pT3 redesign algorithm generated structures which lacked the defined secondary structural elements present in the original 6eh1 starting structure (**Figure 5.4F**). Furthermore, we observed extremely negative average residue Rosetta energies nearing -10 Rosetta energy units (REU) per residue, far away from the -1 to -3 per residue REU rule-of-thumb provided in the Rosetta documentation. These issues at the monomer design level manifested more condensed monomer structures which may have driven the creation of gaps in the VLP models (**Figure 5.4G**). Comparing the initial pT3 redesign script to our proof-of-concept script, we noticed that a bond length constraint was not set in the pT3 redesign script. This difference alone accounted for the loss of secondary structure observed in **Figure 5.4F**, which may have then led to the creation of unphysical protein structures and total failure of the first round of pT3 nanoparticle redesigns. This issue was corrected for the design of future rounds of pT3 VLPs.
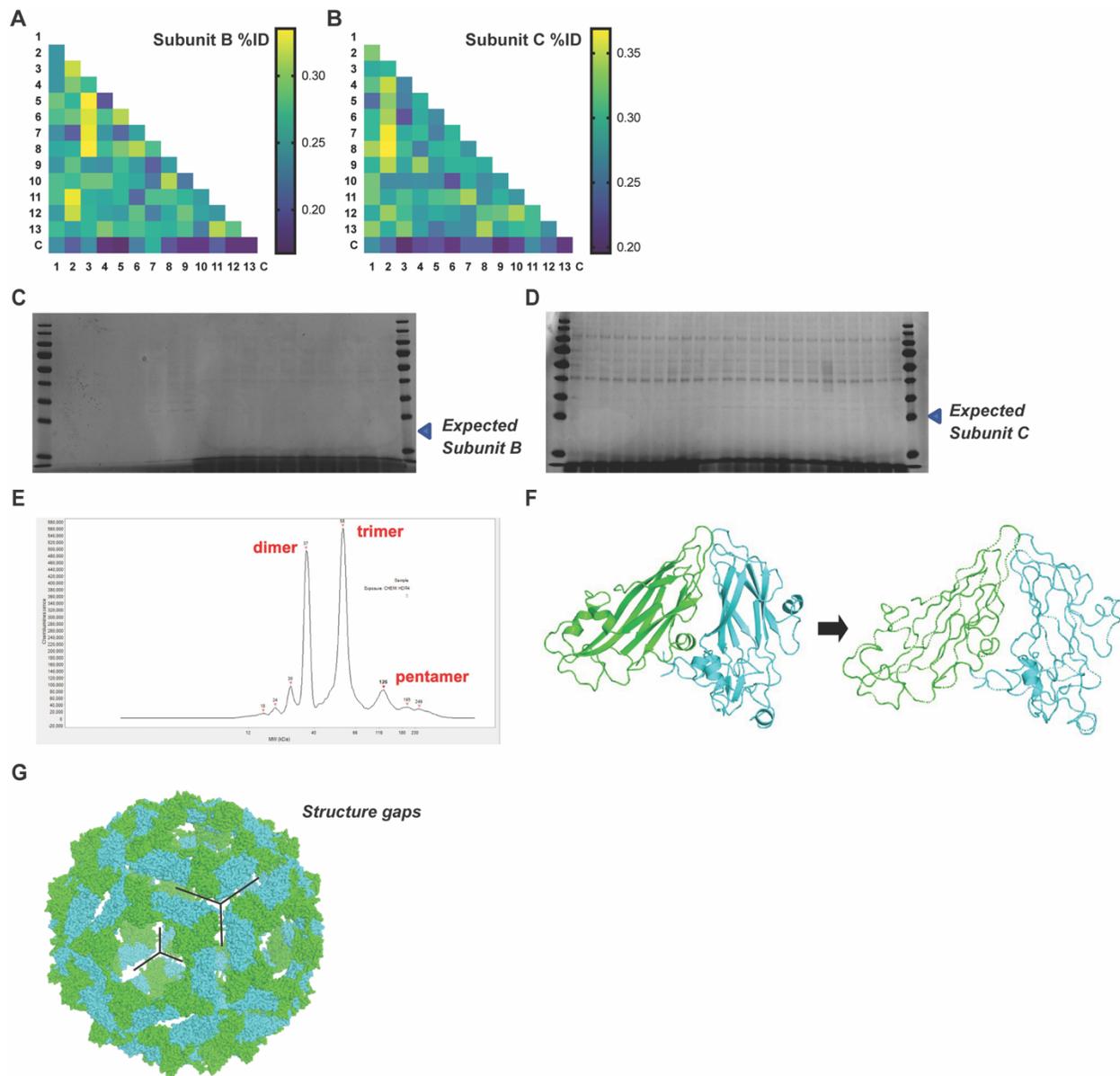
Figure 5.4 Initial redesign of pT3 results in poor expression due to potential multimerization and poor constraint settings

A) Pairwise percent identify of subunit B amino acid sequences from the set of 6eh1 nanoparticle redesigns. B) Pairwise percent identify of subunit C amino acid sequences from the set of 6eh1 nanoparticle redesigns. C) Protein expression observed in initial scout cultures of 6eh1 subunit B after redesign as measured by SDS-PAGE. D) Protein expression observed in initial scout cultures of 6eh1 subunit C after redesign as

measured by SDS-PAGE. E) High molecular weight peaks observed in an anti-histag labeled 6eh1 subunit C intracellular protein sample run on an automated western blotting Jess machine. F) Visualization of 6eh1 subunit B (green) and C (cyan) secondary structure before and after an initial redesign simulation. G) Visualization of representative 6eh1 VLP after an initial redesign simulation.

Targeted truncations and algorithm modifications to address expression challenges

Having diagnosed a couple of likely drivers of poor protein expression in the initial round of pT3 redesigns, we sought to rapidly turn around a new order to test the hypothesis that a complete elimination of homo-multimerization potential and activation of bond length constraints would enable the expression of monomeric B and C components. As shown in **Figure 5.5A**, we truncated the N and C termini of chain B, eliminating residues 61-104 and 235-239 (inclusive), and truncated the C terminus of chain C, eliminating the last 12 residues. The resulting structure and subsequent protein designs derived from it demonstrated no homo-interactions (**Figure 5.5B**) and preserved the original secondary structural elements (**Figure 5.5C**). Using this design algorithm, we ordered linearized plasmids to enable expression testing of 17 new pT3 VLP components from Genscript. Experimental characterization work is ongoing at the time of writing.
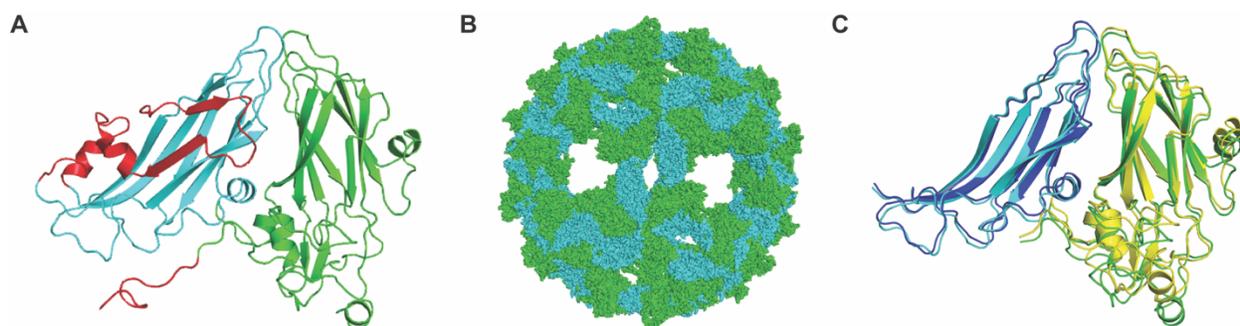
Figure 5.5 Truncations and design algorithm modifications address limitations of the first redesign experiment

A) Truncated regions (red) of 6eh1 subunit B (cyan) and C (green). B) Visualization of truncated 6eh1 VLP structure. C) Overlay of 6eh1 subunit B (cyan/blue) and C (green/yellow), before and after redesign with bond constraints present.

New pT3 capsids improve assembly potential and de-risk expression further

While we were able to eliminate homo-interactions in the second round of designs, we were only able to maintain two of the three hetero-interactions necessary for particle assembly. This issue appears to be attributable to an artifact in the 6eh1 crystal structure, which appears to have a looser connection along the seams equidistant from five-fold symmetry axes that is not present in other pT3 nanoparticle designs (**Figures 5.6A&B**). Initial attempts to correct this issue in 6eh1-derived designs using distance constraints were unsuccessful. Structures derived from truncations of other starting structures, such as PDBIDs 3J2J, 5LK8, 6HBJ, and 6T40, do not have this issue, potentially enabling the preservation of all three hetero-interactions required for successful particle assembly, as shown for 6T40 (**Figure 5.6B&C**). The resulting designs demonstrate the computational feasibility of creating large protein nanoparticles comprised of only hetero-interactions, laying the necessary groundwork for future

experimental validation and iterative development of ultra-manufacturable, highly effective, and thermostable VLP-based vaccines.
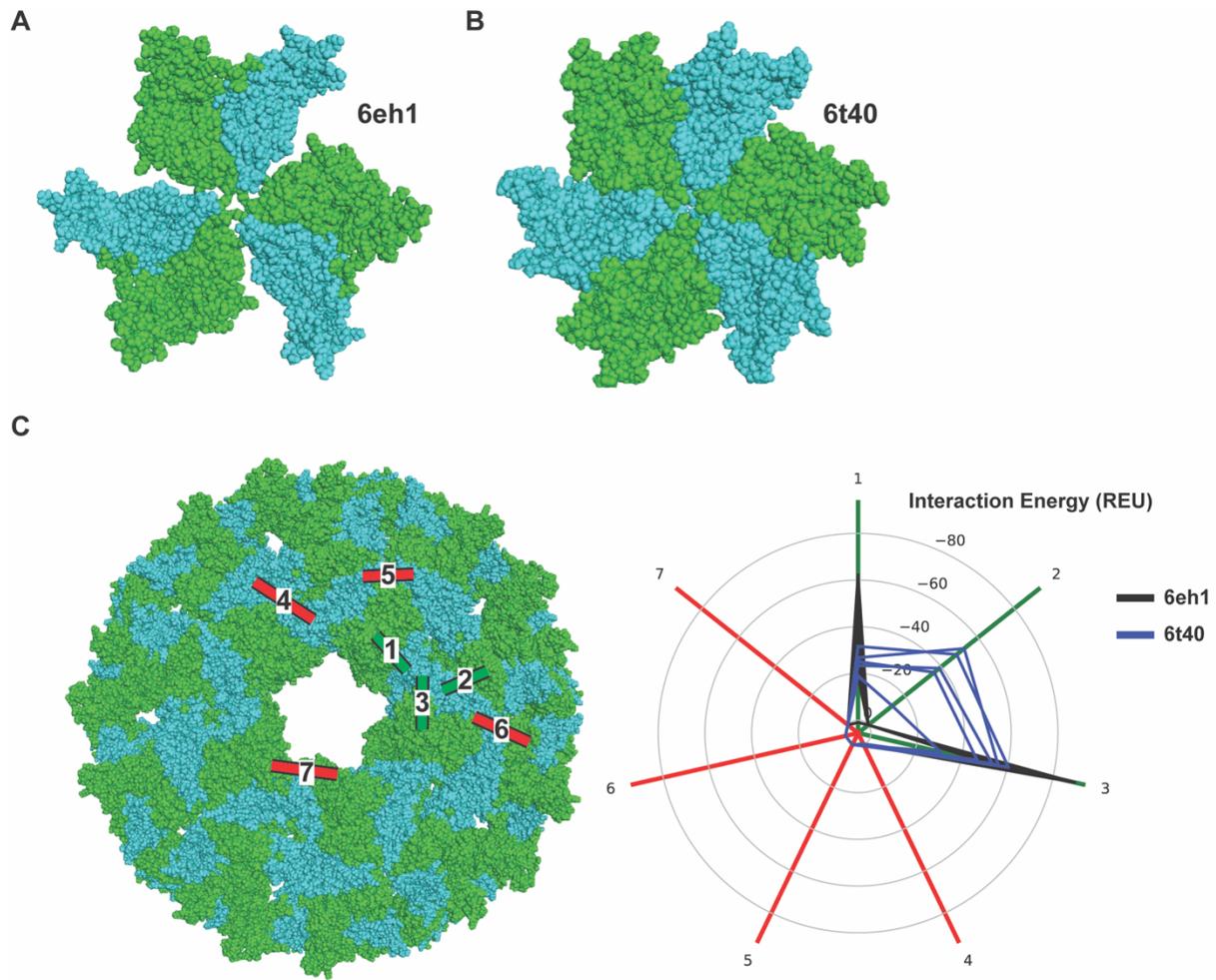
Figure 5.6 New pT3 starting structures enable the creation of designs with all necessary hetero-interactions

A) Representative hetero-hexamer derived from the truncated 6eh1 nanoparticle after redesign. B) Representative hetero-hexamer derived from the truncated 6t40 nanoparticle after redesign. C) Illustration of the 3 hetero-interactions (green) and 4 homo-interactions (red) possible in a pT3 nanoparticle and comparison of interaction energies for redesigned models derived from truncated 6eh1 and 6t40 initial structures.

5.3 Discussion and outlook for future work

Here, we described initial progress towards the computational design of VLPs optimized for expression by *K. phaffii*. To our knowledge, this work represents the first attempt to redesign VLP nanoparticles with pT3 symmetry and the first use of Rosetta to optimize protein designs for secreted expression in *K. phaffii*. To our knowledge, the VLP designs described here are also the first protein VLP designs composed entirely of hetero-multimeric interactions. This project's long-term goal of secreted expression of an *ex vivo* assembling VLP with subsequent purification by nanofiltration is also a novel vaccine production strategy, which could be implemented readily in existing manufacturing facilities, enabling large-scale ultra-low cost production of highly effective and thermostable subunit vaccines.

There are several limitations in the current study which may be addressed by future work. First, while there are many models available for protein design, there are currently no models which can help to predict which protein designs can be expressed by *K. phaffii* and which cannot. Further work will be needed to enable data-driven VLP model down selection and improve the percentage of protein designs which can be readily secreted. Improvements in de novo protein structure prediction could be used as orthogonal validation that protein designs are expected to fold as predicted by Rosetta. Further rule-of-thumbs derived from experience or regression of previous protein expression experiments could further improve the expression success rate. Second, despite many improvements to the computational and experimental workflows, the cycle time from protein design to expression validation is still ~6 weeks. Transient plasmid expression strategies could be developed to reduce this time by another 1-2 weeks,

further accelerating design learning rates. Finally, further work may be necessary to tailor design interfaces for optimal assembly. While scalable tools can provide information regarding interaction strengths, further design algorithm modification to encourage hydrogen bonding or target specific interaction strengths may be necessary to optimize assembly kinetics (Bale et al., 2015; Cannon et al., 2020; Wargacki et al., 2021).

Successful completion of this project could improve future pandemic preparedness and enable ultra low-cost vaccine production for many humanitarian applications. This work may, however, contribute to an even broader trend in therapeutic protein development. Advances in protein structure prediction and protein design have enabled *de novo* design of novel proteins. The integration of protein design and discovery with the manufacturing host of interest, as pursued in this initial work, could enable rapid design of next-gen therapeutics in preferred expression hosts, resulting in lower R&D timelines and manufacturing costs that could ultimately improve patient access to biologics in the developing world. This integrated approach presents exciting opportunities to improve the standard of care in LMICs, providing an early roadmap for improving biologic accessibility by design.

6. Conclusion and outlook

6.1 Summary

In this thesis work, we detailed the need for vaccine design and manufacturing tailored to address unmet medical needs in low- and middle-income countries. We suggested that subunit vaccines designed for optimal production in *K. phaffii* could be a promising solution to this problem and pursued an integrated development approach to develop broadly applicable strategies for rapid, scalable subunit vaccine production in *K. phaffii*. Recognizing that each pathogen derived subunit antigen may present its own unique challenges for production with *K. phaffii*, we pursued the development of broadly applicable strategies to improve subunit vaccine production.

In **Chapters 2&3**, we detail the development and application of strains of *K. phaffii* engineered to eliminate the need for methanol-feeding during bioprocess operation. These strains improved the production of the SARS-CoV-2 derived RBD antigen and enabled rapid scale-up of RBD production in a 1,200 L reactor by partners at the Serum Institute of India. The advantages of methanol-free operation, including improved compatibility with existing facilities, 30% lower heat and oxygen demand, and reduced cell stress, made early methanol-free strains the preferred chassis for production of subsequent RBD-derived subunit components transferred to the Serum Institute of India. Subsequent characterization of generalizability suggests that methanol-free strains may have broadly applicable benefits for many heterologous proteins of interest, although the degree of methanol-free engineering necessary may vary somewhat from protein-to-protein, as demonstrated by our experience with the rotavirus-derived subunit, P[8].

In **Chapter 4**, we detailed the development of a novel modular blending method for cell culture media development and demonstrated that it could be used to identify and optimize the concentration of media components to improve any measurable phenotype of interest. The resulting method achieved a 2x improvement in the secreted titer of P[8] in under 1 month and identified carbon and lipid metabolism as particularly important for achieving optimal production of heterologous protein in *K. phaffii*. The resulting production medium appears broadly applicable for production of many heterologous proteins of interest and the method is generalizable to other phenotypes of interest, such as improving the rate of biomass accumulation, and potentially even for other alternative production hosts.

Finally, in **Chapter 5**, we presented initial work towards the design and production of a broadly applicable and highly manufacturable thermostable VLP, which could be used for immunogenic display of any antigen of interest. Unlike in previous chapters, where sequence modifications were somewhat constrained by the need to generate antigen-specific immune responses against the pathogen from which the antigen was derived, VLP protein sequence design actually benefits from being loosely constrained, as the ability to generate many different VLP candidates enables optimization of protein design for secretability, potential elimination of immunogenic epitopes, and rapid development of alternative VLPs if immune responses to a previous VLP arise. Recognizing this freedom, we developed computational and experimental workflows to enable the design and rapid screening of VLP candidates tailored for optimal production in *K. phaffii*. The resulting pT3 VLP designs may be the first icosahedral protein nanoparticles to contain only heteromeric interactions. Initial proof-of-concept work suggests that pT3 VLP

designs may eventually be secreted at high titer to enable *ex vivo* nanoparticle

assembly and simple downstream purification.

6.2 Remaining challenges and future directions

Improving global access to biologic drugs, such as subunit vaccines, will require

reductions in manufacturing cost-of-goods and improvements in volumetric productivity.

Ultimately, as evident in much of this thesis work, volumetric productivity is the result of

a balance between product design and manufacturing capabilities.

In the first part of this work, we manipulated transcription factor expression levels

and improved media design to yield broadly applicable improvements to manufacturing

capabilities. There are a few potential ways to extend or build off this initial work. First,

transcription factors are key regulars of many cellular processes, but only a small

minority of transcription factors have been explored in *K. phaffii*. Controlled screening of

novel TFs, as done in Chapter 3, is a labor-intensive process which is somewhat limited

in the scope of accessible transcriptional landscapes it can explore. The development of

multiplex plasmid libraries to enable broad modulation of TF networks could serve as an

interesting discovery platform for further host engineering efforts. If one considers that

there are ~160 TFs in *K. phaffii*, a cloning workflow which overexpresses random

combinations of four TFs at a time could enable the creation and screening of over 600

million unique transcriptional states in a single assay. Paired with a good screening

method, such a discovery platform could accelerate the discovery of novel beneficial

host genome modifications. Second, media development work highlighted the

importance of optimal carbon source feeding strategy and lipid supplementation for

heterologous protein production. A deeper and more targeted exploration of these pathways via further blending experiments or metabolic modulation could yield further generalizable benefits for the production of heterologous proteins with *K. phaffii*.

In addition to ongoing efforts to expand manufacturing capabilities, it has become increasingly clear that tailoring product design for optimal production in a manufacturing host of interest is a promising method for improving volumetric productivity; however, much remains to be learned about the characteristics common to proteins that can be highly secreted by *K. phaffii* and about the acceptable limits to protein modification in therapeutic protein design, which could be constrained by factors such as anti-drug immunogenicity. Ultimately, the ability to rapidly design and screen many novel protein designs may enable the creation of data sets which may help answer these questions, improving the design of new protein structures and the selection and modification of product candidates derived largely from existing proteins. Thus, while improvements will continue to be made to the underlying manufacturing capabilities of *K. phaffii*, improved understanding of current limitations and allowable product modifications will enable selection and design of products tailored to take advantage of the many strengths of this expression system.

6.3 Implications

The thesis work presented here explored what an optimal vaccine platform for LMICs might look like and addressed several challenges towards the realization of this goal. While some challenges remain, we hope the technologies developed here may contribute to improved access to high quality vaccines and biologics in LMICs, and that

future biologics development efforts may consider closer integration of discovery and manufacturing innovations with the goal of further improving global access to vaccines and other beneficial biologic drugs.

7. References

Acharya, K. P., Ghimire, T. R., & Subramanya, S. H. (2021). Access to and equitable distribution of COVID-19 vaccine in low-income countries. *Npj Vaccines*, *6*(1), 2–4. https://doi.org/10.1038/s41541-021-00323-6

Ahmad, M., Hirz, M., Pichler, H., & Schwab, H. (2014). Protein expression in Pichia pastoris: recent achievements and perspectives for heterologous protein production. https://doi.org/10.1007/s00253-014-5732-5

Amani, S., & Naeem, A. (2013). Understanding protein folding from globular to amyloid state: Aggregation: Darker side of protein. *Process Biochemistry*, *48*(11), 1651–1664. https://doi.org/10.1016/j.procbio.2013.08.011

Apte-Deshpande, A., Rewanwar, S., Kotwal, P., Raiker, V. A., & Padmanabhan, S. (2009). Efficient expression and secretion of recombinant human growth hormone in the methylotrophic yeast Pichia pastoris : potential applications for other proteins, *205*, 197–205. https://doi.org/10.1042/BA20090179

Ardila-leal, L. D., Alvarado-ramírez, M. F., Guti, I. S., & Pedroza-rodríguez, A. M. (2020). Heliyon Low-cost media statistical design for laccase rPOXA 1B production in P . pastoris, *6*(March). https://doi.org/10.1016/j.heliyon.2020.e03852

Baden, L. R., El Sahly, H. M., Essink, B., Kotloff, K., Frey, S., Novak, R., … Zaks, T. (2021). Efficacy and Safety of the mRNA-1273 SARS-CoV-2 Vaccine. *New England Journal of Medicine*, *384*(5), 403–416. https://doi.org/10.1056/nejmoa2035389

Bale, J. B., Gonen, S., Liu, Y., Sheffler, W., Ellis, D., Thomas, C., … Baker, D. (2016). Accurate design of megadalton-scale two-component icosahedral protein complexes. *Science*, *353*(6297), 389–395. https://doi.org/10.5061/dryad.8c65s

Bale, J. B., Park, R. U., Liu, Y., Gonen, S., Gonen, T., Cascio, D., … Baker, D. (2015). Structure of a designed tetrahedral protein assembly variant engineered to have improved soluble expression. *The Protein Society*, *24*, 1695–1701. https://doi.org/10.1002/pro.2748

Baumann, K., Adelantado, N., Lang, C., Mattanovich, D., & Ferrer, P. (2011). Protein trafficking , ergosterol biosynthesis and membrane physics impact recombinant protein secretion in Pichia pastoris.

Biedermann, A. M., Gengaro, I. R., Rodriguez-Aponte, S. A., Love, K. R., & Love, J. C. (2021). Modular development enables rapid design of media for alternative hosts. *BioRxiv*, 1–29.

Biedermann, A. M., Gengaro, I. R., Rodriguez-Aponte, S. A., Love, K. R., & Love, J. C. (2022). Modular development enables rapid design of media for alternative hosts.

Biotechnology and Bioengineering, *119*(1), 59–71. https://doi.org/10.1002/bit.27947

Brady, J. R., & Love, J. C. (2021). Alternative hosts as the missing link for equitable therapeutic protein production. *Nature Biotechnology*, *39*(4), 404–407. https://doi.org/10.1038/s41587-021-00884-w

Brady, J. R., Whittaker, C. A., Tan, M. C., Kristensen, D. L., Ma, D., Dalvie, N. C., … Love, J. C. (2020). Comparative genome-scale analysis of Pichia pastoris variants informs selection of an optimal base strain. *Biotechnology and Bioengineering*, *117*(2), 543–555. https://doi.org/10.1002/bit.27209

Brühlmann, D., Sokolov, M., Butté, A., Sauer, M., Hemberger, J., Souquet, J., … Jordan, M. (2017). Parallel experimental design and multivariate analysis provides efficient screening of cell culture media supplements to improve biosimilar product quality. *Biotechnology and Bioengineering*, *114*(7), 1448–1458. https://doi.org/10.1002/bit.26269

Burgard, J., Valli, M., Graf, A. B., Gasser, B., & Mattanovich, D. (2017). Biomarkers allow detection of nutrient limitations and respective supplementation for elimination in Pichia pastoris fed - batch cultures. *Microbial Cell Factories*, 1–17. https://doi.org/10.1186/s12934-017-0730-9

Butler, M., Huzel, N., Barnab, N., Gray, T., & Bajno, L. (1999). Linoleic acid improves the robustness of cells in agitated cultures, 27–36.

Cannon, K. A., Park, R. U., Boyken, S. E., Nattermann, U., Yi, S., Baker, D., … Yeates, T. O. (2020). Design and structure of two new protein cages illustrate successes and ongoing challenges in protein engineering, (December 2019), 919–929. https://doi.org/10.1002/pro.3802

Carnicer, M., Baumann, K., Töplitz, I., Sánchez-, F., Mattanovich, D., Ferrer, P., & Albiol, J. (2009). Macromolecular and elemental composition analysis and extracellular metabolite balances of Pichia pastoris growing at different oxygen levels, *14*, 1–14. https://doi.org/10.1186/1475-2859-8-65

Chang, C. H., Hsiung, H. A., Hong, K. L., & Huang, C. T. (2018). Enhancing the efficiency of the Pichia pastoris AOX1 promoter via the synthetic positive feedback circuit of transcription factor Mxr1. *BMC Biotechnology*, *18*(1), 1–10. https://doi.org/10.1186/s12896-018-0492-4

Chen, W.-H., Hotez, P. J., & Bottazzi, M. E. (2020). Potential for developing a SARS-CoV receptor-binding domain (RBD) recombinant protein as a heterologous human vaccine against coronavirus infectious disease (COVID)-19. *Human Vaccines & Immunotherapeutics*, *16*(6), 1239–1242. https://doi.org/10.1080/21645515.2020.1740560

Clincke, M.-F., Guedon, E., Yen, F. T., Ea, L., Universite, N., Ogier, V., … Goergen, J.-L. (n.d.). Effect of Surfactant Pluronic F-68 on CHO Cell Growth , Metabolism ,

Production , and Glycosylation of Human Recombinant IFN- c in Mild Operating Conditions, 181–190. https://doi.org/10.1002/btpr.503

Coffman, J., Brower, M., Connell-Crowley, L., Deldari, S., Farid, S. S., Horowski, B., … Shultz, J. (2021). A common framework for integrated and continuous biomanufacturing. *Biotechnology and Bioengineering*, *118*(4), 1721–1735. https://doi.org/10.1002/bit.27690

Cohen, A. A., Gnanapragasam, P. N. P., Lee, Y. E., Hoffman, P. R., Ou, S., Kakutani, L. M., … Bjorkman, P. J. (2021). Mosaic nanoparticles elicit cross-reactive immune responses to zoonotic coronaviruses in mice. *Science (New York, N.Y.)*. https://doi.org/10.1126/science.abf6840

Coleman, E. (2020). Establishment of a Novel Pichia Pastoris Host Production Platform by.

Combs, G. F. (2012). *The vitamins: Fundamental aspects in nutrition and health*. San Diego: Elsevier Academic Press.

Cooper, R. A. (1978). Influence of Increased Membrane Cholesterol on Membrane Fluidity and Cell Function in Human Red Blood Cells. *Journal of Supramolecular Structure*.

Crowell, L. E., Crowell, L. E., Raymond, A., St, H. E., Engineering, C., Doyle, P. S., & Crowell, L. E. (2020). Accelerating process development for biologics on an automated , pharmacy-scale manufacturing system by by.

Crowell, L. E., Lu, A. E., Love, K. R., Stockdale, A., Timmick, S. M., Wu, D., … Love, J. C. (2018, November). On-demand manufacturing of clinical-quality biopharmaceuticals. *Nature Biotechnology*. Nature Publishing Group. https://doi.org/10.1038/nbt.4262

Dai, L., Zheng, T., Xu, K., Han, Y., Xu, L., Huang, E., … Gao, G. F. (2020). A Universal Design of Betacoronavirus Vaccines against COVID-19, MERS, and SARS. *Cell*, *182*(3), 722-733.e11. https://doi.org/10.1016/j.cell.2020.06.035

Dalvie, N. C., Biedermann, A. M., Rodriguez-Aponte, S. A., Naranjo, C. A., Rao, H. D., Rajurkar, M. P., … Love, J. C. (2021a). Scalable, methanol-free manufacturing of the SARS-CoV-2 receptor-binding domain in engineered Komagataella phaffii. *Biotechnology and Bioengineering*. https://doi.org/10.1002/bit.27979

Dalvie, N. C., Biedermann, A. M., Rodriguez-Aponte, S. A., Naranjo, C. A., Rao, H. D., Rajurkar, M. P., … Love, J. C. (2021b). Scalable, methanol-free manufacturing of the SARS-CoV-2 receptor binding domain in engineered Komagataella phaffii. *BioRxiv : The Preprint Server for Biology*, 1–18. https://doi.org/10.1101/2021.04.15.440035

Dalvie, N. C., Brady, J. R., Crowell, L. E., Tracey, M. K., Biedermann, A. M., Kaur, K., …

Love, J. C. (2020). Molecular engineering improves antigen quality and enables integrated manufacturing of a trivalent subunit vaccine candidate for rotavirus. *BioRx*, 1–51.

Dalvie, N. C., Leal, J., Whittaker, C. A., Yang, Y., Brady, J. R., Love, K. R., & Christopher Love, J. (2020). Host-Informed Expression of CRISPR Guide RNA for Genomic Engineering in Komagataella phaffii. *ACS Synthetic Biology*, *9*(1), 26–35. https://doi.org/10.1021/acssynbio.9b00372

Dalvie, N. C., Leal, J., Whittaker, C. A., Yang, Y., Brady, J. R., Love, K. R., & Love, J. C. (2019). Host-Informed Expression of CRISPR Guide RNA for Genomic Engineering in *Komagataella phaffii. ACS Synthetic Biology*, acssynbio.9b00372. https://doi.org/10.1021/acssynbio.9b00372

Dalvie, N. C., Lorgeree, T., Biedermann, A. M., Love, K. R., & Love, J. C. (2021). Simplified Gene Knockout by CRISPR-Cas9-Induced Homologous Recombination. *ACS Synthetic Biology*. https://doi.org/10.1021/acssynbio.1c00194

Dalvie, N. C., Rodriguez-aponte, S. A., Hartwell, B. L., Tostanoski, L. H., Biedermann, A. M., Crowell, L. E., … Carter, L. (2021). Engineered SARS-CoV-2 receptor binding domain improves manufacturability in yeast and immunogenicity in mice, *118*(38). https://doi.org/10.1073/pnas.2106845118/-/DCSupplemental.Published

Degreif, D., Cucu, B., Budin, I., Thiel, G., & Bertl, A. (2019). Lipid determinants of endocytosis and exocytosis in budding yeast. *BBA - Molecular and Cell Biology of Lipids*, *1864*(7), 1005–1016. https://doi.org/10.1016/j.bbalip.2019.03.007

Der, B. S., Kluwe, C., Miklos, A. E., Jacak, R., Lyskov, S., Gray, J. J., … Kuhlman, B. (2013). Alternative Computational Protocols for Supercharging Protein Surfaces for Reversible Unfolding and Retention of Stability. *PLoS ONE*, *8*(5). https://doi.org/10.1371/journal.pone.0064363

Duke Global Health Innovation Center, _. (2021). Mapping COVID-19 vaccine pre-purchases across the globe. Retrieved from https://launchandscalefaster.org/COVID-19

Farinha, I., Araújo, D., & Freitas, F. (2019). Optimization of medium composition for production of chitin-glucan complex and mannose-containing polysaccharides by the yeast Komagataella pastoris. *Journal of Biotechnology*, *303*(September 2018), 30–36. https://doi.org/10.1016/j.jbiotec.2019.07.007

Forman, R., Shah, S., Jeurissen, P., Jit, M., & Mossialos, E. (2021). COVID-19 vaccine challenges: What have we learned so far and what remains to be done? *Health Policy*, *125*(5), 553–567. https://doi.org/10.1016/j.healthpol.2021.03.013

Fraser, R. Z., Shitut, M., Agrawal, P., Mendes, O., & Klapholz, S. (2018). Safety Evaluation of Soy Leghemoglobin Protein Preparation Derived From Pichia pastoris, Intended for Use as a Flavor Catalyst in Plant-Based Meat. *International*

Journal of Toxicology, *37*(3), 241–262. https://doi.org/10.1177/1091581818766318

FSA Panel on Additives and Products or Substances used in Animal Feed (FEEDAP). (2011). Scientific Opinion on safety and efficacy of choline chloride as a feed additive for all animal species. *EFSA Journal*, *9*(9), 2353.

Gagnon, M., Hiller, G., Luan, Y. T., Kittredge, A., Defelice, J., & Drapeau, D. (2011). High-End pH-controlled delivery of glucose effectively suppresses lactate accumulation in CHO Fed-batch cultures. *Biotechnology and Bioengineering*, *108*(6), 1328–1337. https://doi.org/10.1002/bit.23072

Galbraith, S. C., Bhatia, H., Liu, H., & Yoon, S. (2018). Media formulation optimization: current and future opportunities. *Current Opinion in Chemical Engineering*, *22*, 42–47. https://doi.org/10.1016/j.coche.2018.08.004

Ghosalkar, A., Sahai, V., & Srivastava, A. (2008). Optimization of chemically defined medium for recombinant Pichia pastoris for biomass production. *Bioresource Technology*, *99*(16), 7906–7910.

Ghosh, P. K. (2017). Similar Biologics : Global Opportunities and Issues, *19*(4), 552–596.

Guebre-Xabier, M., Patel, N., Tian, J. H., Zhou, B., Maciejewski, S., Lam, K., … Smith, G. (2020). NVX-CoV2373 vaccine protects cynomolgus macaque upper and lower airways against SARS-CoV-2 challenge. *Vaccine*. https://doi.org/10.1016/j.vaccine.2020.10.064

Guo, C., Huang, Y., Zheng, H., Tang, L., He, J., Xiang, L., … Jiang, H. (2012). Secretion and activity of antimicrobial peptide cecropin D expressed in Pichia pastoris. *Experimental and Therapeutic Medicine*, 1063–1068. https://doi.org/10.3892/etm.2012.719

Haque, A., & Pant, A. B. (2020). Efforts at COVID-19 vaccine development: Challenges and successes. *Vaccines*, *8*(4), 1–16. https://doi.org/10.3390/vaccines8040739

Hartner, F. S., & Glieder, A. (2006). Regulation of methanol utilisation pathway genes in yeasts, *21*, 1–21. https://doi.org/10.1186/1475-2859-5-39

Heath, P. T., Galiza, E. P., Baxter, D. N., Boffito, M., Browne, D., Burns, F., … Toback, S. (2021). Safety and Efficacy of NVX-CoV2373 Covid-19 Vaccine. *New England Journal of Medicine*, *385*(13), 1172–1183. https://doi.org/10.1056/nejmoa2107659

Inan, M., & Meagher, M. M. (2001). Non-repressing carbon sources for alcohol oxidase (AOX1) Promoter of Pichia pastoris. *Journal of Bioscience and Bioengineering*, *92*(6), 585–589.

Invitrogen Corporation. (2002). Pichia Fermentation Process Guidelines Overview Overview , continued. *Progress in Botany*, *67*, 1–11.

Jacak, R., Leaver-fay, A., & Kuhlman, B. (2011). Computational protein design with explicit consideration of surface hydrophobic patches, (November), 825–838. https://doi.org/10.1002/prot.23241

Jiang, H., Horwitz, A. A., Wright, C., Tai, A., Znameroski, E. A., Tsegaye, Y., … Love, J. C. (2019). Challenging the workhorse: Comparative analysis of eukaryotic micro-organisms for expressing monoclonal antibodies. *Biotechnology and Bioengineering*, *116*(6), 1449–1462. https://doi.org/10.1002/bit.26951

Jordan, M., Voisard, D., Berthoud, A., & Tercier, L. (2013). Cell culture medium improvement by rigorous shuffling of components using media blending, 31–40. https://doi.org/10.1007/s10616-012-9462-1

Kallel, H., & Kamen, A. A. (2015). Large-scale adenovirus and poxvirus-vectored vaccine manufacturing to enable clinical trials. *Biotechnology Journal*, *10*(5), 741–747. https://doi.org/10.1002/biot.201400390

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., … Yamanishi, Y. (2008). KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, *36*(SUPPL. 1), 480–484. https://doi.org/10.1093/nar/gkm882

Kaushik, N., Lamminmäki, U., Khanna, N., & Batra, G. (2020). Enhanced cell density cultivation and rapid expression-screening of recombinant Pichia pastoris clones in microscale, 1–11. https://doi.org/10.1038/s41598-020-63995-5

Kelley, B. (2020). Developing therapeutic monoclonal antibodies at pandemic pace. *Nature Biotechnology*. https://doi.org/10.1038/s41587-020-0512-5

Kelly, H. G., Kent, S. J., & Wheatley, A. K. (2019). Immunological basis for enhanced immunity of nanoparticle vaccines. *Expert Review of Vaccines*, *18*(3), 269–280. https://doi.org/10.1080/14760584.2019.1578216

Kennedy, M., & Krouse, D. (1999). Strategies for improving fermentation medium performance: a review. *Journal of Industrial Microbiology and Biotechnology*, (23), 456–475.

Kim, J. H., Marks, F., & Clemens, J. D. (2021). Looking beyond COVID-19 vaccine phase 3 trials. *Nature Medicine*, *27*(2), 205–211. https://doi.org/10.1038/s41591-021-01230-y

King, N. P., Bale, J. B., Sheffler, W., Mcnamara, D. E., Gonen, S., Gonen, T., & Yeates, T. O. (2014). Accurate design of co-assembling multi-component protein nanomaterials. *Nature*. https://doi.org/10.1038/nature13404

Kleanthous, H., Silverman, J. M., Makar, K. W., Yoon, I.-K., Jackson, N., & Vaughn, D. W. (2021). Scientific rationale for developing potent RBD-based vaccines targeting COVID-19. *Npj Vaccines*, *6*(1), 1–10. https://doi.org/10.1038/s41541-021-00393-6

Kuryatov, A., Mukherjee, J., & Lindstrom, J. (2013). Chemical Chaperones Exceed the Chaperone Effects of RIC-3 in Promoting Assembly of Functional a 7 AChRs. *PLoS ONE*, *8*(4), 1–11. https://doi.org/10.1371/journal.pone.0062246

Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., & Aprahamian, M. (2020). Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nature Methods*, *17*, 665–680. https://doi.org/10.1038/s41592-020-0848-2

Lin-Cereghino, G. P., Godfrey, L., de la Cruz, B. J., Johnson, S., Khuongsathiene, S., Tolstorukov, I., … Cregg, J. M. (2006). Mxr1p, a Key Regulator of the Methanol Utilization Pathway and Peroxisomal Genes in Pichia pastoris. *Molecular and Cellular Biology*, *26*(3), 883–897. https://doi.org/10.1128/mcb.26.3.883-897.2006

Lin, N.-X., He, R.-Z., Xu, Y., & Yu, X.-W. (2021). Oxidative stress tolerance contributes to heterologous protein production in Pichia pastoris. *Biotechnology for Biofuels 2021 14:1*, *14*(1), 1–13. https://doi.org/10.1186/S13068-021-02013-W

Loebrich, S., Clark, E., Ladd, K., Takahashi, S., Brousseau, A., Kitchener, S., … Ryll, T. (2019). Comprehensive manipulation of glycosylation profiles across development scales. *MAbs*, *11*(2), 335–349. https://doi.org/10.1080/19420862.2018.1527665

Love, J. C., Love, K. R., & Barone, P. W. (2012). Enabling global access to high-quality biopharmaceuticals. *Current Opinion in Chemical Engineering*, *2*(4), 383–390. https://doi.org/10.1016/j.coche.2013.09.002

Love, K. R., Dalvie, N. C., & Love, J. C. (n.d.). The yeast stands alone : the future of protein biologic production. *Current Opinion in Biotechnology*, *53*, 50–58. https://doi.org/10.1016/j.copbio.2017.12.010

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2, 1–21. https://doi.org/10.1186/s13059-014-0550-8

Lu, T. L., Pugach, O., Somerville, R., Rosenberg, S. A., Kochenderfer, J. N., Better, M., & Feldman, S. A. (2016). A Rapid Cell Expansion Process for Production of Engineered Autologous CAR-T Cell Therapies. *Human Gene Therapy Methods*, *27*(6), 209–219. https://doi.org/10.1089/hgtb.2016.120

Mahammad, S., & Parmryd, I. (2015). Cholesterol depletion using methyl-β-cyclodextrin. In *Methods in membrane lipids* (pp. 91–102).

Marsalek, L., Puxbaum, V., Buchetics, M., Mattanovich, D., & Gasser, B. (2019). Disruption of vacuolar protein sorting components of the HOPS complex leads to enhanced secretion of recombinant proteins in Pichia pastoris. *Microbial Cell Factories*, 1–16. https://doi.org/10.1186/s12934-019-1155-4

Massinga Loembé, M., & Nkengasong, J. N. (2021). COVID-19 vaccine access in

Africa: Global distribution, vaccine platforms, and challenges ahead. *Immunity*, *54*(7), 1353–1362. https://doi.org/10.1016/j.immuni.2021.06.017

Mathieu, E., Ritchie, H., Ortiz-Ospina, E., Roser, M., Hasell, J., Appel, C., … Rodés-Guirao, L. (2021). A global database of COVID-19 vaccinations. *Nature Human Behaviour*, *5*(7), 947–953. https://doi.org/10.1038/s41562-021-01122-8

Matthews, C. B., Kuo, A., Love, K. R., & Love, J. C. (2017a). Development of a general defined medium for Pichia pastoris, (July), 103–113. https://doi.org/10.1002/bit.26440

Matthews, C. B., Kuo, A., Love, K. R., & Love, J. C. (2017b). Development of a general defined medium for Pichia pastoris, (July), 103–113. https://doi.org/10.1002/bit.26440

Matthews, C. B., Wright, C., Kuo, A., Colant, N., Westoby, M., & Love, J. C. (2017). Reexamining opportunities for therapeutic protein production in eukaryotic microorganisms, (May), 2432–2444. https://doi.org/10.1002/bit.26378

McCarthy, D. J., Chen, Y., & Smyth, G. K. (2012). Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, *40*(10), 4288–4297. https://doi.org/10.1093/nar/gks042

McGillicuddy, N., Floris, P., Albrecht, S., & Bones, J. (2018). Examining the sources of variability in cell culture media used for biopharmaceutical production. *Biotechnology Letters*, *40*(1), 5–21. https://doi.org/10.1007/s10529-017-2437-8

Mellet, J., & Pepper, M. S. (2021). A covid-19 vaccine: Big strides come with big challenges. *Vaccines*, *9*(1), 1–14. https://doi.org/10.3390/vaccines9010039

Mohmad-Saberi, S. E., Hashim, Y. Z. H. Y., Mel, M., Amid, A., Ahmad-Raus, R., & Packeer-Mohamed, V. (2013). Metabolomics profiling of extracellular metabolites in CHO-K1 cells cultured in different types of growth media. *Cytotechnology*, *65*(4), 577–586. https://doi.org/10.1007/s10616-012-9508-4

Moleirinho, M. G., Silva, R. J. S., Alves, P. M., Manuel, J. T., Peixoto, C., Moleirinho, M. G., … Manuel, J. T. (2020). Current challenges in biotherapeutic particles manufacturing. *Expert Opinion on Biological Therapy*, *20*(5), 451–465. https://doi.org/10.1080/14712598.2020.1693541

Montiel-Garcia, D., Santoyo-Rivera, N., Ho, P., Carrillo-Tripp, M., Brooks, C. L., Johnson, J. E., & Reddy, V. S. (2021). VIPERdb v3.0: A structure-based data analytics platform for viral capsids. *Nucleic Acids Research*, *49*(D1), D809–D816. https://doi.org/10.1093/nar/gkaa1096

Moser, J. W., Prielhofer, R., Gerner, S. M., Graf, A. B., Wilson, I. B. H., Mattanovich, D., & Dragosits, M. (2017). Implications of evolutionary engineering for growth and recombinant protein production in methanol - based growth media in the yeast

Pichia pastoris. *Microbial Cell Factories*, 1–16. https://doi.org/10.1186/s12934-017-0661-5

Österlund, T., Bordel, S., & Nielsen, J. (2015). Controllability analysis of transcriptional regulatory networks reveals circular control patterns among transcription factors. *Integrative Biology (United Kingdom)*, *7*(5), 560–568. https://doi.org/10.1039/c4ib00247d

Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, *14*(4), 417–419. https://doi.org/10.1038/nmeth.4197

Pereira, S., Kildegaard, H. F., & Andersen, M. R. (2018). Impact of CHO Metabolism on Cell Growth and Protein Production: An Overview of Toxic and Inhibiting Metabolites and Nutrients. *Biotechnology Journal*, *13*(3), 1–13. https://doi.org/10.1002/biot.201700499

Pollet, J., Chen, W.-H., Versteeg, L., Keegan, B., Zhan, B., Wei, J., … Bottazzi, M. E. (2020). SARS-CoV-2 RBD219-N1C1: A Yeast-Expressed SARS-CoV-2 Recombinant Receptor-Binding Domain Candidate Vaccine Stimulates Virus Neutralizing Antibodies and T-cell Immunity in Mice. *BioRxiv : The Preprint Server for Biology*, 2020.11.04.367359. https://doi.org/10.1101/2020.11.04.367359

Potvin, G., Ahmad, A., & Zhang, Z. (2012). Bioprocess engineering aspects of heterologous protein production in Pichia pastoris: A review. *Biochemical Engineering Journal*, *64*, 91–105. https://doi.org/10.1016/j.bej.2010.07.017

Pulliam, J. R. C., Schalkwyk, C. van, Govender, N., Gottberg, A. von, Cohen, C., Groome, M. J., … Moultrie, H. (2021). Increased risk of SARS-CoV-2 reinfection associated with emergence of the Omicron variant in South Africa. *MedRxiv*, 2021.11.11.21266068. Retrieved from https://www.medrxiv.org/content/10.1101/2021.11.11.21266068v2%0Ahttps://www.medrxiv.org/content/10.1101/2021.11.11.21266068v2.abstract

Reddington, S. C., & Howarth, M. (2015, December). Secrets of a covalent interaction for biomaterials and biotechnology: SpyTag and SpyCatcher. *Current Opinion in Chemical Biology*. Elsevier Ltd. https://doi.org/10.1016/j.cbpa.2015.10.002

Ritacco, Frank V; Yongqi Wu, A. K. (2018). Cell Culture Media for Recombinant Protein Expression in Chinese Hamster Ovary (CHO) Cells : History , Key Components , and Optimization Strategies. https://doi.org/10.1002/btpr.2706

Ritacco, F. V., Wu, Y., & Khetan, A. (2018). Cell culture media for recombinant protein expression in Chinese hamster ovary (CHO) cells: History, key components, and optimization strategies. *Biotechnology Progress*, *34*(6), 1407–1426. https://doi.org/10.1002/btpr.2706

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2009). edgeR: A Bioconductor

package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Rodrigues, M. E., Costa, A. R., Henriques, M., Azeredo, J., & Oliveira, R. (2012). Comparison of commercial serum-free media for CHO-K1 cell growth and monoclonal antibody production. *International Journal of Pharmaceutics*, *437*(1–2), 303–305. https://doi.org/10.1016/j.ijpharm.2012.08.002

Rouiller, Y., Périlleux, A., Collet, N., Jordan, M., Stettler, M., & Broly, H. (2013). A high-throughput media design approach for high performance mammalian fed-batch cultures, (June), 501–511.

Routledge, S. J., Hewitt, C. J., Bora, N., & Bill, R. M. (2011). Antifoam addition to shake flask cultures of recombinant Pichia pastoris increases yield, 1–11.

S. Ahi, Y., S. Bangari, D., & K. Mittal, S. (2011). Adenoviral Vector Immunity: Its Implications and Circumvention Strategies. *Current Gene Therapy*, *11*(4), 307–320. https://doi.org/10.2174/156652311796150372

Sahu, U., Krishna Rao, K., & Rangarajan, P. N. (2014). Trm1p, a Zn(II)2Cys6-type transcription factor, is essential for the transcriptional activation of genes of methanol utilization pathway, in Pichia pastoris. *Biochemical and Biophysical Research Communications*, *451*(1), 158–164. https://doi.org/10.1016/j.bbrc.2014.07.094

Sankar, K., Krystek, S. R., Carl, S. M., Day, T., & Maier, J. K. X. (2018). AggScore: Prediction of aggregation-prone regions in proteins based on the distribution of surface patches. *Proteins: Structure, Function and Bioinformatics*, *86*(11), 1147–1156. https://doi.org/10.1002/prot.25594

Schnellbaecher, A., Binder, D., Bellmaine, S., & Zimmer, A. (2019). Vitamins in cell culture media: Stability and stabilization strategies. *Biotechnology and Bioengineering*, *116*(6), 1537–1555. https://doi.org/10.1002/bit.26942

Sell, T. K., Gastfriend, D., Watson, M., Watson, C., Richardson, L., Cicero, A., … Connell, N. (2021). Building the global vaccine manufacturing capacity needed to respond to pandemics. *Vaccine*, *39*(12), 1667–1669. https://doi.org/10.1016/j.vaccine.2021.02.017

Sharma, O., Sultan, A. A., Ding, H., & Triggle, C. R. (2020). A Review of the Progress and Challenges of Developing a Vaccine for COVID-19. *Frontiers in Immunology*, *11*(December 2019), 1–17. https://doi.org/10.3389/fimmu.2020.585354

Shi, L., Wang, J., Wang, X., Zhang, Y., Song, Z., Cai, M., & Zhou, X. (2019). Transcriptome and metabolome analyses reveal global behaviour of a genetically engineered methanol-independent Pichia pastoris strain. *Process Biochemistry*, *76*(October 2018), 46–54. https://doi.org/10.1016/j.procbio.2018.10.014

Soares, H., Antunes, F., Marinho, H. S., & Real, C. (2014). Redox Biology Hydrogen peroxide sensing , signaling and regulation of transcription factors, *2*, 535–562. https://doi.org/10.1016/j.redox.2014.02.006

Sompayrac, L. (2012). *How the Immune System Works.*

Soneson, C., Love, M. I., & Robinson, M. D. (2016). Differential analyses for RNA-seq: Transcript-level estimates improve gene-level inferences [version 2; referees: 2 approved]. *F1000Research*, *4*. https://doi.org/10.12688/F1000RESEARCH.7563.2

Soontorngun, N. (2017). Reprogramming of nonfermentative metabolism by stress-responsive transcription factors in the yeast Saccharomyces cerevisiae. *Current Genetics*, *63*(1), 1–7. https://doi.org/10.1007/s00294-016-0609-z

Sorkun, M. C., Khetan, A., & Er, S. (2019). AqSolDB , a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds, *2019*(July), 1–8. https://doi.org/10.1038/s41597-019-0151-1

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., … Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles.

Tavasoli, T., Arjmand, S., Omid, S., Siadat, R., & Shojaosadati, S. A. (2017). Enhancement of Alpha 1-antitrypsin Production in Pichia pastoris by Designing and Optimizing Medium Using Elemental Analysis, *15*(4). https://doi.org/10.15171/ijb.1808

Tian, J.-H., Patel, N., Haupt, R., Zhou, H., Weston, S., Lague, J., … Smith, G. (2020). Title: SARS-CoV-2 spike glycoprotein vaccine candidate NVX-CoV2373 elicits 1 immunogenicity in baboons and protection in mice 2 3. *BioRxiv*, 2020.06.29.178509. https://doi.org/10.1101/2020.06.29.178509

Tinari, S., & Riva, C. (2021). Covid-19: Whatever happened to the Novavax vaccine? *Bmj*, n2965. https://doi.org/10.1136/bmj.n2965

Totaro, D., Radoman, B., Schmelzer, B., Rothbauer, M., Steiger, M. G., Mayr, T., … Mattanovich, D. (2021). Microscale Perfusion-Based Cultivation for Pichia pastoris Clone Screening Enables Accelerated and Optimized Recombinant Protein Production Processes, *2000215*, 1–8. https://doi.org/10.1002/biot.202000215

Turcotte, B., Liang, X. B., Robert, F., & Soontorngun, N. (2010). Transcriptional regulation of nonfermentable carbon utilization in budding yeast. *FEMS Yeast Research*, *10*(1), 2–13. https://doi.org/10.1111/j.1567-1364.2009.00555.x

Twarock, R., & Luque, A. (2019). Structural puzzles in virology solved with an overarching icosahedral design principle. *Nature Communications*, 1–9. https://doi.org/10.1038/s41467-019-12367-3

Uppala, J. K., Gani, A. R., & Ramaiah, K. V. A. (2017). Chemical chaperone , TUDCA unlike PBA , mitigates protein aggregation efficiently and resists ER and non-ER stress induced HepG2 cell death. *Scientific*, *1*(January), 1–13. https://doi.org/10.1038/s41598-017-03940-1

Vanz, A., Lünsdorf, H., Adnan, A., Nimtz, M., Gurramkonda, C., Khanna, N., & Rinas, U. (2012). Physiological response of Pichia pastoris GS115 to methanol-induced high level production of the Hepatitis B surface antigen: catabolic adaptation, stress responses, and autophagic processes. *Microbial Cell Factories 2012 11:1*, *11*(1), 1–11. https://doi.org/10.1186/1475-2859-11-103

Vellanki, R. N., Potumarthi, R., & Mangamoori, L. N. (2009). Constitutive Expression and Optimization of Nutrients for Streptokinase Production by Pichia pastoris Using Statistical Methods, 25–40. https://doi.org/10.1007/s12010-008-8315-z

Veneziano, R., Moyer, T. J., Stone, M. B., Wamhoff, E. C., Read, B. J., Mukherjee, S., … Bathe, M. (2020). Role of nanoscale antigen organization on B-cell activation probed using DNA origami. *Nature Nanotechnology*, *15*(8), 716–723. https://doi.org/10.1038/s41565-020-0719-0

Villadsen, J. (2015). Redox Balances and Consistency Check of Experiments. In *Fundamental Bioengineering* (pp. 17–38).

Vogl, T., Glieder, A., Sturmberger, L., Fauland, P. C., Hyden, P., Fischer, J. E., … Geier, M. (2018). Methanol independent induction in Pichia pastoris by simple derepressed overexpression of single transcription factors, (October 2017), 1037–1050. https://doi.org/10.1002/bit.26529

Vogl, T., Sturmberger, L., Fauland, P. C., Hyden, P., Fischer, J. E., Schmid, C., … Glieder, A. (2018). Methanol independent induction in *Pichia pastoris* by simple derepressed overexpression of single transcription factors. *Biotechnology and Bioengineering*, *115*(4), 1037–1050. https://doi.org/10.1002/bit.26529

Wakayama, K., Yamaguchi, S., Takeuchi, A., Mizumura, T., Ozawa, S., Tomizuka, N., … Nakagawa, T. (2016). Regulation of intracellular formaldehyde toxicity during methanol metabolism of the methylotrophic yeast Pichia methanolica. *Journal of Bioscience and Bioengineering*, *122*(5), 545–549. https://doi.org/10.1016/j.jbiosc.2016.03.022

Walls, A. C., Fiala, B., Schäfer, A., Wrenn, S., Pham, M. N., Murphy, M., … King, N. P. (2020). Elicitation of Potent Neutralizing Antibody Responses by Designed Protein Nanoparticle Vaccines for SARS-CoV-2. *Cell*, *183*(5), 1367-1382.e17. https://doi.org/10.1016/j.cell.2020.10.043

Wang, J., Wang, X., Shi, L., Qi, F., Zhang, P., Zhang, Y., … Cai, M. (2017). Methanol-Independent Protein Expression by AOX1 Promoter with trans-Acting Elements Engineering and Glucose-Glycerol-Shift Induction in Pichia pastoris. *Scientific Reports*, *7*(December 2016), 1–12. https://doi.org/10.1038/srep41850

Wang, X., Wang, Q., Wang, J., Bai, P., Shi, L., Shen, W., … Cai, M. (2016). Mit1 transcription factor mediates methanol signaling and regulates the alcohol oxidase 1 (AOX1) promoter in pichia pastoris. *Journal of Biological Chemistry*, *291*(12), 6245–6261. https://doi.org/10.1074/jbc.M115.692053

Wargacki, A. J., Wörner, T. P., Waterbeemd, M. Van De, Ellis, D., Heck, A. J. R., & King, N. P. (2021). Complete and cooperative in vitro assembly of computationally designed self-assembling protein nanomaterials. *Nature Communications*, 1–14. https://doi.org/10.1038/s41467-021-21251-y

Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., … Hughes, T. R. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, *158*(6), 1431–1443. https://doi.org/10.1016/j.cell.2014.08.009

Xu, C., Huang, P., Ravichandran, R., Yi, S., Davis, T. N., Gonen, T., & Neil, P. (2016). Design of a hyperstable 60-subunit protein icosahedron. *Nature*, *535*(7610), 136–139. https://doi.org/10.1038/nature18010

Yamamoto, T., & Ishihara, K. (n.d.). Stability of Glutathione in Solution. *Developments in Food Engineering*, 209–211.

Yang, Y., Zheng, Y., Wang, P., Li, X., Zhan, C., & Linhardt, R. J. (2020). Characterization and application of a putative transcription factor (SUT2) in Pichia pastoris. *Molecular Genetics and Genomics*, *1*(0123456789). https://doi.org/10.1007/s00438-020-01697-3

Zakharova, E. M., & Minashina, I. K. (2015). Review of Multidimensional Optimization Methods. *Mathematical Models, Computational Methods*, *60*(6), 625–636. https://doi.org/10.1134/S1064226915060194

Zhang, H., Wang, H., & Liu, M. (2013). Rational development of a serum-free medium and fed-batch process for a GS-CHO cell line expressing recombinant antibody, 363–378. https://doi.org/10.1007/s10616-012-9488-4

Zhang, P., Zhang, W., Zhou, X., Bai, P., Cregg, J. M., & Zhang, Y. (2010). Catabolite Repression of Aox in Pichia pastoris is dependent on hexose transporter PpHxt1 and pexophagy. *Applied and Environmental Microbiology*, *76*(18), 6108–6118. https://doi.org/10.1128/AEM.00607-10

Zhang, W., Sinha, J., & Meagher, M. M. (2006). Glycerophosphate as a phosphorus source in a defined medium for Pichia pastoris fermentation, 139–144. https://doi.org/10.1007/s00253-005-0238-9

A. Appendix
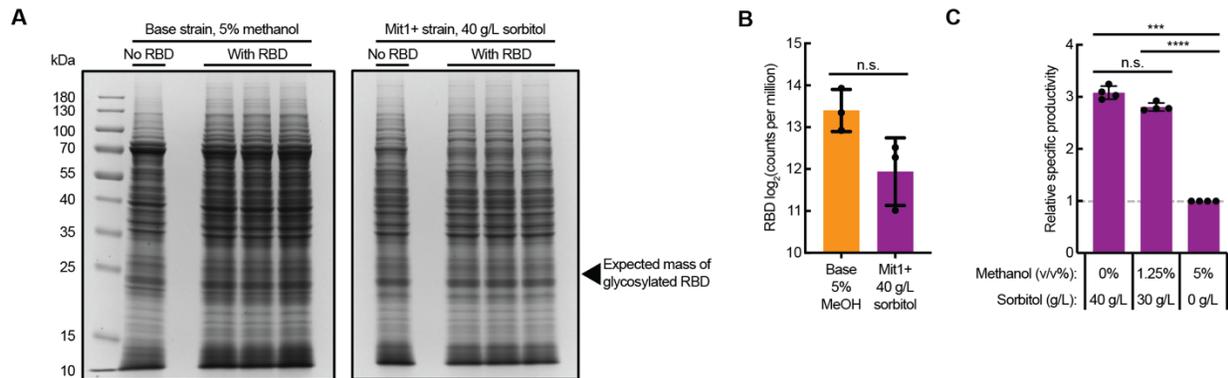
Figure A.1 Further comparison of methanol-free and methanol-fed cultivation conditions for the production of SARS-CoV-2 RBD

A) SDS-PAGE of intracellular protein after cultivation of the base strain and the *mit1+*

strain, with and without the recombinant RBD gene. The base strain was cultivated with

5% methanol feed, and the *mit1+* strain was cultivated with 40 g/L sorbitol feed. B)

Abundance of the recombinant RBD transcript in the base strain and the *mit1+* strain,

measured by RNA sequencing. Significance was determined by unpaired t-test. C)

Relative specific productivity of the *mit1+* strain cultivated with different feed conditions.

Four biological replicates were normalized to the corresponding replicate in the 5%

methanol condition. Significance was determined by ratio paired t-test.

****p<0.00001, ***p<0.0001, n.s. indicates p > 0.05
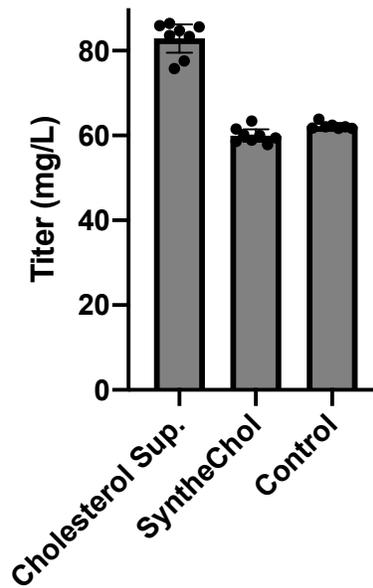
Figure A.2 Synthetic cholesterol supplementation alone does not elicit a productivity enhancement comparable to the multicomponent cholesterol supplement
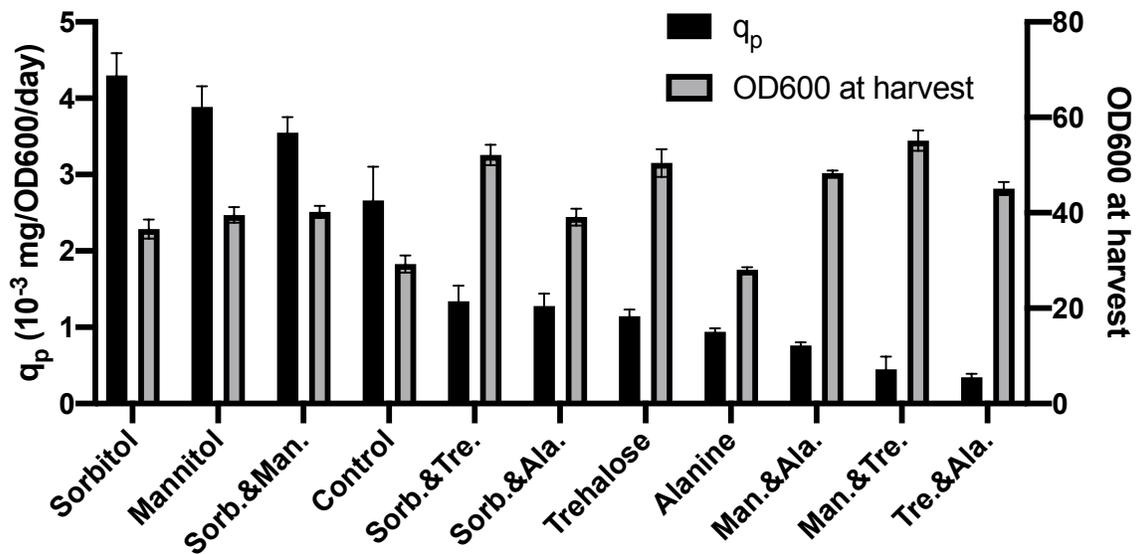
Figure A.3 Specific productivity and OD600 at harvest for co-feed supplement screen

Derivation of a novel pT3 icosahedral symmetry file

Motivation

Computation design of self-assembling nanoparticles with Rosetta may require the derivation of novel symmetry files. While Rosetta provides a high-level overview of symmetry file construction in the original paper, the online tutorials and existing symmetry file construction tools within Rosetta only cover and work for simple use cases, such as the symmetric rotation of a monomer about an axis. The construction of novel symmetry files for particle design with icosahedral symmetries is non-trivial, requiring careful reading of the original Rosetta symmetry paper and considerable trial and error. Without such a file, redesign of most symmetric particles of interest is likely to be computationally infeasible. Here, I provide a more detailed overview of the custom construction of Rosetta symmetry files to aid researchers faced with this challenge in the future. This discussion should enable the construction new symmetry files with correct geometry, although there remains some ambiguity regarding best practices for non-geometric definitions in the symmetry file.

Overview of a Rosetta symmetry file

A Rosetta symmetry file contains information which defines how a larger molecular structure can be constructed from a minimum set of unique subunits. Given a pdb file containing this minimum set of subunits, Rosetta uses the symmetry file to place copies of the subunits on the faces of a polyhedron, aligned along specific axes that correspond to symmetry groups. For example, in a 2-component nanoparticle, a 120 subunit structure can be constructed by identifying the 60 unique basis vectors which

define how groups of 2 subunits are laid out on the particle surface. By inspection, one can see that pT3 symmetry is a pseudo-symmetry consisting of individual 5-fold symmetry groups. For a nanoparticle with pT3 symmetry, this can be easily done by identifying the vectors that connect the center of each pentagonal face of a truncated icosahedron to the vertices (there are 60 total vertices).
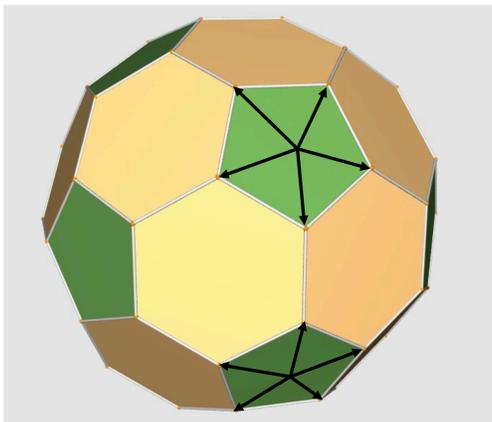
Figure A.4 Unit vectors begin at the center of each face and point toward the vertices of the truncated icosahedron

Most of the Rosetta symmetry file handles these basic geometry definitions and understanding the meaning of these geometric definitions is most important for defining new symmetry files—other parameters can be adapted empirically with minimal trial and error. For example, there is an energy definition line which tells Rosetta how to estimate the total energy of the structure, based on the interactions of one subunit with other symmetric copies of that subunit.

Deriving fold tree and geometry of pT3 symmetry definition

Rosetta's symmetry file uses a tree of virtual coordinates to define a symmetry file. When constructing this tree, it is helpful to think of the tree as containing layers of

branches. The tree contains a single root. It then branches into two separate trees, one for component A and one for component B. Each tree in pT3 symmetry is geometrically identical. There are two layers containing unit vectors that point to the center of each pentagonal face (two layers is a convention I copied from IPD symmetry files—it is not clear why two layers are needed mathematically). Each of these branches that points to the center of a pentagonal face then connects to virtual coordinates which are unit vectors connecting the center of the pentagonal face to its 5 vertices. Each of these final branches are connected to the center of mass of the subunit which is symmetrically rotated about the tree. Details related to jump groups and degrees of freedom were copied empirically from IPD symmetry files (reasoning and essentiality of these is unclear, but they basically group the jumps to each face and from the center of each face to the vertices).
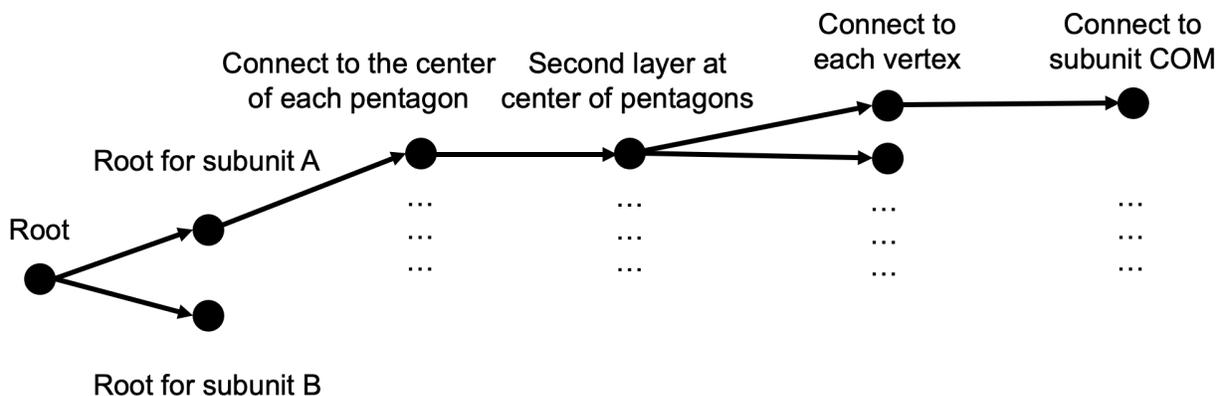
Figure A.5 Fold tree for a multi-component symmetric nanoparticle

The details of geometric derivation are contained in the following code segments. First, Antiprism software installed to the command line is used to obtain the geometric definition of a truncated icosahedron. test_plato_shapes.py then runs code in PlatoShapes.py and SymmetryFile.py to output a symmetry file named test.txt.

125

Additional methods could be added to incorporate additional components using the same geometric definitions in the future.

Using Antiprism to generate a .obj file with initial vectors (cmd line)

```
1.    $> /usr/local/bin/off2obj U25 > trunc_icos.obj
```

Where the U25 parameter references an existing object:
http://www.antiprism.com/examples/150_named_models/520_uniform/index.html

test_plato_shapes.py

```
1.  #!/usr/bin/env python
2.
3.  from PlatoShapes import TruncatedIcosahedron
4.  from SymmetryFile import SymmetryFile
5.
6.  foo = TruncatedIcosahedron("trunc_icos.obj")
7.
8.  bar = SymmetryFile(foo,"test.txt","foo")
9.  bar.write_sym_file()
```

SymmetryFile.py

```python
1.  #!/usr/bin/env python
2.
3.  import numpy as np
4.
5.  class SymmetryFile(object):
6.      def __init__(self,shape,outfile_name,name="foo"):
7.          self.shape = shape
8.          self.outfile = open(outfile_name,"wt")
9.          self.name = name
10.         self.group_xy_plain_by_x()
11.         print(self.face_groups)
12.         return
13.
14.     def group_xy_plain_by_x(self):
15.         data = self.shape.xy_origin
16.         self.face_groups = dict()
17.         for i in data:
18.             if tuple(i[:3]) not in self.face_groups.keys():
19.                 self.face_groups[tuple(i[:3])] = []
20.             self.face_groups[tuple(i[:3])].append(i)
21.         for key in self.face_groups.keys():
22.             self.face_groups[key] = np.array(self.face_groups[key])
23.         return
24.
25.     def write_header(self):
26.         self.outfile.write("symmetry_name "+self.name+"\n\nsubunits
    60\n\nnumber_of_interfaces 59\n\n\n")
27.         return
28.
29.     def write_subunitA_energy(self):
30.         this_str = "E = 60*BTA0_0+"
31.         for i,key in enumerate(self.face_groups.keys()):
32.             for j, point in enumerate(self.face_groups[key]):
33.                 if (i==0)&(j==0):
34.                     continue
35.                 this_str+="30*(BTA0_0:BTA"+str(i)+"_"+str(j)+")+"
36.         this_str = this_str[:-1]
37.         this_str+="\n"
38.         self.outfile.write(this_str)
39.         return
40.
41.     def write_subunitB_energy(self):
42.         this_str = "E = 60*BTB0_0+"
43.         for i,key in enumerate(self.face_groups.keys()):
44.             for j, point in enumerate(self.face_groups[key]):
45.                 if (i==0)&(j==0):
46.                     continue
```

```python
47.             this_str+="30*(BTB0_0:BTB"+str(i)+"_"+str(j)+")+"
48.         this_str = this_str[:-1]
49.         this_str+="\n"
50.         self.outfile.write(this_str)
51.         return
52.
53.     def write_anchor_residue(self):
54.         self.outfile.write("anchor_residue COM\n\n")
55.         return
56.
57.     def write_intermediate_layer_coordinates(self,prefix):
58.         for i,key in enumerate(self.face_groups.keys()):
59.             point = self.face_groups[key][0]
60.             self.outfile.write("xyz "+prefix+str(i)+" "+
61.                 str(point[0])+","+str(point[1])+","+str(point[2])+" "+
62.                 str(point[3])+","+str(point[4])+","+str(point[5])+" "+
63.                 str(point[6])+","+str(point[7])+","+str(point[8])+"\n")
64.         self.outfile.write("\n")
65.         return
66.
67.     def write_last_virtual_layer_coordinates(self,prefix):
68.         for i,key in enumerate(self.face_groups.keys()):
69.             for j, point in enumerate(self.face_groups[key]):
70.                 self.outfile.write("xyz "+prefix+str(i)+"_"+str(j)+" "+
71.                     str(point[0])+","+str(point[1])+","+str(point[2])+" "+
72.                     str(point[3])+","+str(point[4])+","+str(point[5])+" "+
73.                     str(point[6])+","+str(point[7])+","+str(point[8])+"\n")
74.         self.outfile.write("\n")
75.         return
76.
77.     def write_virtual_coords(self):
78.         self.outfile.write("virtual_coordinates_start\n\n")
79.         self.outfile.write("xyz ROOT +1,+0,+0 +0,+1,+0 0,0,0\n\n")
80.         self.outfile.write("xyz RA +1,+0,+0 +0,+1,+0 0,0,0\n\n")
81.         self.outfile.write("xyz RB +1,+0,+0 +0,+1,+0 0,0,0\n\n")
82.         self.write_intermediate_layer_coordinates("CTA")
83.         self.write_intermediate_layer_coordinates("CTB")
84.         self.write_intermediate_layer_coordinates("PTA")
85.         self.write_intermediate_layer_coordinates("PTB")
86.         self.write_last_virtual_layer_coordinates("BTA")
87.         self.write_last_virtual_layer_coordinates("BTB")
88.         self.outfile.write("virtual_coordinates_stop\n\n")
89.
90.     def connect_root_to_first_layer(self,jump_counter,root_name,prefix):
91.         jump_list = []
92.         for i,key in enumerate(self.face_groups.keys()):
93.             jmp_name = "JMP"+str(jump_counter)
94.             jump_counter+=1
95.             jump_list.append(jmp_name)
```

```python
96.
97.          self.outfile.write("connect_virtual "+jmp_name+" "+root_name
98.                +" "+prefix+str(i)+"\n")
99.       self.outfile.write("\n")
100.             return([jump_counter,jump_list])
101.
102.        def connect_intermediate_layers(self,jump_counter,prefix1,prefix2):
103.            jump_list = []
104.            for i,key in enumerate(self.face_groups.keys()):
105.               jmp_name = "JMP"+str(jump_counter)
106.               jump_counter+=1
107.               jump_list.append(jmp_name)
108.
109.               self.outfile.write("connect_virtual "+jmp_name+" "+prefix1+str(i)
110.                    +" "+prefix2+str(i)+"\n")
111.            self.outfile.write("\n")
112.            return([jump_counter,jump_list])
113.
114.         def
    connect_intermediate_to_last_virtual_layer(self,jump_counter,prefix1,prefix2):
115.            jump_list = []
116.            for i,key in enumerate(self.face_groups.keys()):
117.                for j, point in enumerate(self.face_groups[key]):
118.                    jmp_name = "JMP"+str(jump_counter)
119.                    jump_counter+=1
120.                    jump_list.append(jmp_name)
121.
122.                    self.outfile.write("connect_virtual "+jmp_name+" "+prefix1+str(i)
123.                        +" "+prefix2+str(i)+"_"+str(j)+"\n")
124.            self.outfile.write("\n")
125.            return([jump_counter,jump_list])
126.
127.         def connect_last_virtual_to_subunit(self,jump_counter,prefix,subunit_prefix):
128.            jump_list = []
129.            for i,key in enumerate(self.face_groups.keys()):
130.                for j, point in enumerate(self.face_groups[key]):
131.                    jmp_name = "JMP"+str(jump_counter)
132.                    jump_counter+=1
133.                    jump_list.append(jmp_name)
134.
135.                    self.outfile.write("connect_virtual "+jmp_name+" "+prefix+str(i)
136.                        +"_"+str(j)+" "+subunit_prefix+"\n")
137.            self.outfile.write("\n")
138.            return([jump_counter,jump_list])
139.
140.         def write_connections(self):
141.            jump_counter = 0
142.            jmp_groups = []
143.
```

```
144.            self.outfile.write("connect_virtual JMP0 ROOT RA\n")
145.            jump_counter+=1
146.            self.outfile.write("connect_virtual JMP1 ROOT RB\n")
147.            jump_counter+=1
148.
149.            this_output = self.connect_root_to_first_layer(jump_counter,"RA","CTA")
150.            jump_counter = this_output[0]
151.            #jmp_groups.append(this_output[1])
152.
153.            this_output = self.connect_root_to_first_layer(jump_counter,"RB","CTB")
154.            jump_counter = this_output[0]
155.            #jmp_groups.append(this_output[1])
156.
157.            this_output = self.connect_intermediate_layers(jump_counter,"CTA","PTA")
158.            jump_counter = this_output[0]
159.            jmp_groups.append(this_output[1])
160.
161.            this_output = self.connect_intermediate_layers(jump_counter,"CTB","PTB")
162.            jump_counter = this_output[0]
163.            jmp_groups.append(this_output[1])
164.
165.            this_output =
    self.connect_intermediate_to_last_virtual_layer(jump_counter,"PTA","BTA")
166.            jump_counter = this_output[0]
167.            #jmp_groups.append(this_output[1])
168.
169.            this_output =
    self.connect_intermediate_to_last_virtual_layer(jump_counter,"PTB","BTB")
170.            jump_counter = this_output[0]
171.            #jmp_groups.append(this_output[1])
172.
173.            this_output =
    self.connect_last_virtual_to_subunit(jump_counter,"BTA","SUBUNIT B COM")
174.            jump_counter = this_output[0]
175.            jmp_groups.append(this_output[1])
176.
177.            this_output =
    self.connect_last_virtual_to_subunit(jump_counter,"BTB","SUBUNIT C COM")
178.            jump_counter = this_output[0]
179.            jmp_groups.append(this_output[1])
180.
181.            return(jmp_groups)
182.
183.        def write_dofs(self):
184.            self.outfile.write("\nset_dof JMP0 x angle_x\n\n")
185.            self.outfile.write("\nset_dof JMP1 x angle_x\n\n")
186.            return
187.
188.        def write_jump_groups(self,jmp_groups):
```

```python
189.            for i, g in enumerate(jmp_groups):
190.                this_str = ""
191.                for j in g:
192.                    this_str+=j+" "
193.                this_str+="\n"
194.                self.outfile.write("set_jump_group JGR"+str(chr(65+i))+" "+this_str)
195.            return
196.
197.        def write_sym_file(self):
198.            self.write_header()
199.            self.write_subunitA_energy()
200.            self.write_subunitB_energy()
201.            self.write_anchor_residue()
202.            self.write_virtual_coords()
203.            jmp_groups = self.write_connections()
204.            self.write_dofs()
205.            self.write_jump_groups(jmp_groups)
206.
207.            self.outfile.close()
208.            return
209.
```

PlatoShapes.py

```python
1.  #!/usr/bin/env python
2.
3.  import numpy as np
4.  from mpl_toolkits import mplot3d
5.  from mpl_toolkits.mplot3d import Axes3D
6.  from mpl_toolkits.mplot3d.art3d import Poly3DCollection
7.  import matplotlib.pyplot as plt
8.
9.
10. class TruncatedIcosahedron(object):
11.     def __init__(self,obj_filename):
12.         data = open(obj_filename).readlines()
13.         vdata = [i for i in data if i.split()[0]=='v']
14.         fdata = [i for i in data if i.split()[0]=='f']
15.         self.vertices = [[float(i.split()[1]),float(i.split()[2]),float(i.split()[3])]
16.                          for i in vdata]
17.         self.vertices = np.array(self.vertices)
18.         hex_faces = []
19.         pent_faces = []
20.         for i in fdata:
21.             entry = i.split()[1:]
22.             if len(entry)==5:
23.                 pent_faces.append([int(entry[0])-1,
24.                             int(entry[1])-1,
25.                             int(entry[2])-1,
26.                             int(entry[3])-1,
27.                             int(entry[4])-1])
28.             if len(entry)==6:
29.                 hex_faces.append([int(entry[0])-1,
30.                             int(entry[1])-1,
31.                             int(entry[2])-1,
32.                             int(entry[3])-1,
33.                             int(entry[4])-1,
34.                             int(entry[5])-1])
35.         self.hex_faces = np.array(hex_faces)
36.         self.pent_faces = np.array(pent_faces)
37.
38.         # Find pentagon centroids
39.         self.pcentroids = np.array([np.mean(self.vertices[i],axis=0) for i in pent_faces])
40.         self.get_xy_planes()
41.         return
42.
43.     def get_xy_planes(self):
44.         xy_origin = [] # deriving Rosetta vectors
45.         for i, pcent in enumerate(self.pcentroids):
46.             for j, vert in enumerate(self.pent_faces[i]):
47.                 x = pcent
```

```python
48.             y = np.array(x-self.vertices[vert])
49.             x = x/np.linalg.norm(x)
50.             y = y/np.linalg.norm(y)
51.             xy_origin.append([x[0],x[1],x[2],
52.                             y[0],y[1],y[2],
53.                             0,0,0])
54.         self.xy_origin = np.array(xy_origin)
55.         return
56.
57.     def get_closest_points(self,origin,hexagon):
58.         distances = np.array([np.linalg.norm(origin-self.vertices[i]) for i in hexagon])
59.         this_hex = hexagon[np.argsort(distances)]
60.         return(this_hex)
61.
62.     def plot_faces(self):
63.         fig = plt.figure(figsize = (10, 7))
64.         ax = Axes3D(fig)
65.
66.         this_v = self.vertices[self.hex_faces.flatten()]
67.         colors = ['b','g','r','c','m','y','darkorange','cornflowerblue',
68.                 'mediumpurple','maroon','salmon','darkseagreen',
69.                 'pink','peru','darkkhaki','slategrey','k','bisque',
70.                 'gold','teal','plum']
71.         for i in np.arange(0,len(this_v),6):
72.             this_poly = [list(zip(list(this_v[i:i+6,0]),
73.                             list(this_v[i:i+6,1]),
74.                             list(this_v[i:i+6,2])))]
75.             idx = int((i+1)/6)
76.             if i==0:
77.                 idx=0
78.             ax.add_collection3d(Poly3DCollection(this_poly,color=colors[idx]))
79.         ax.set_xlim([-4,4])
80.         ax.set_ylim([-4,4])
81.         ax.set_zlim([-4,4])
82.         plt.show()
83.         return
84.
85.     def plot_points(self):
86.         fig = plt.figure(figsize = (10, 7))
87.         ax = Axes3D(fig)
88.
89.         this_v = self.vertices[self.hex_faces.flatten()]
90.         ax.scatter(this_v[:,0],this_v[:,1],this_v[:,2],color='b')
91.         ax.scatter(self.pcentroids[:,0],self.pcentroids[:,1],self.pcentroids[:,2],color='r')
92.         ax.set_xlim([-4,4])
93.         ax.set_ylim([-4,4])
94.         ax.set_zlim([-4,4])
95.         plt.show()
96.         return
```

97.

test_symfile.py

```python
1.  #!/usr/bin/env python
2.
3.  from pyrosetta import *
4.  init()
5.
6.  import sys
7.
8.  ref_name = sys.argv[1]
9.  pdb_name = sys.argv[2]
10. sym_name = sys.argv[3]
11.
12. ref_pose = pose_from_pdb(ref_name)
13.
14. pose = pose_from_pdb(pdb_name)
15. sym = rosetta.protocols.symmetry.SetupForSymmetryMover(sym_name)
16.
17. pose2 = pose.clone()
18. sym.apply(pose2)
19.
20. rmsd = pyrosetta.rosetta.core.simple_metrics.metrics.RMSDMetric()
21. rmsd.set_comparison_pose(ref_pose)
22.
23. this_rmsd = rosetta.core.scoring.CA_rmsd(ref_pose,pose2)
24. print(this_rmsd)
25.
26. pmm = PyMOLMover()
27. pmm.apply(pose)
28. pmm.apply(pose2)
```

l5bee_2.sym

```
1.  symmetry_name foo
2.
3.  subunits 60
4.
5.  number_of_interfaces 59
6.
7.
8.  E =
    60*BTA0_0+30*(BTA0_0:BTA0_1)+30*(BTA0_0:BTA0_2)+30*(BTA0_0:BTA0_3)+30*(BTA0_0:BTA0_4)+30*(BT
    A0_0:BTA1_0)+30*(BTA0_0:BTA1_1)+30*(BTA0_0:BTA1_2)+30*(BTA0_0:BTA1_3)+30*(BTA0_0:BTA1_4)+30*
    (BTA0_0:BTA2_0)+30*(BTA0_0:BTA2_1)+30*(BTA0_0:BTA2_2)+30*(BTA0_0:BTA2_3)+30*(BTA0_0:BTA2_4)+
    30*(BTA0_0:BTA3_0)+30*(BTA0_0:BTA3_1)+30*(BTA0_0:BTA3_2)+30*(BTA0_0:BTA3_3)+30*(BTA0_0:BTA3_
    4)+30*(BTA0_0:BTA4_0)+30*(BTA0_0:BTA4_1)+30*(BTA0_0:BTA4_2)+30*(BTA0_0:BTA4_3)+30*(BTA0_0:BT
    A4_4)+30*(BTA0_0:BTA5_0)+30*(BTA0_0:BTA5_1)+30*(BTA0_0:BTA5_2)+30*(BTA0_0:BTA5_3)+30*(BTA0_0
    :BTA5_4)+30*(BTA0_0:BTA6_0)+30*(BTA0_0:BTA6_1)+30*(BTA0_0:BTA6_2)+30*(BTA0_0:BTA6_3)+30*(BTA
    0_0:BTA6_4)+30*(BTA0_0:BTA7_0)+30*(BTA0_0:BTA7_1)+30*(BTA0_0:BTA7_2)+30*(BTA0_0:BTA7_3)+30*(
    BTA0_0:BTA7_4)+30*(BTA0_0:BTA8_0)+30*(BTA0_0:BTA8_1)+30*(BTA0_0:BTA8_2)+30*(BTA0_0:BTA8_3)+3
    0*(BTA0_0:BTA8_4)+30*(BTA0_0:BTA9_0)+30*(BTA0_0:BTA9_1)+30*(BTA0_0:BTA9_2)+30*(BTA0_0:BTA9_3
    )+30*(BTA0_0:BTA9_4)+30*(BTA0_0:BTA10_0)+30*(BTA0_0:BTA10_1)+30*(BTA0_0:BTA10_2)+30*(BTA0_0:
    BTA10_3)+30*(BTA0_0:BTA10_4)+30*(BTA0_0:BTA11_0)+30*(BTA0_0:BTA11_1)+30*(BTA0_0:BTA11_2)+30*
    (BTA0_0:BTA11_3)+30*(BTA0_0:BTA11_4)
9.  E =
    60*BTB0_0+30*(BTB0_0:BTB0_1)+30*(BTB0_0:BTB0_2)+30*(BTB0_0:BTB0_3)+30*(BTB0_0:BTB0_4)+30*(BT
    B0_0:BTB1_0)+30*(BTB0_0:BTB1_1)+30*(BTB0_0:BTB1_2)+30*(BTB0_0:BTB1_3)+30*(BTB0_0:BTB1_4)+30*
    (BTB0_0:BTB2_0)+30*(BTB0_0:BTB2_1)+30*(BTB0_0:BTB2_2)+30*(BTB0_0:BTB2_3)+30*(BTB0_0:BTB2_4)+
    30*(BTB0_0:BTB3_0)+30*(BTB0_0:BTB3_1)+30*(BTB0_0:BTB3_2)+30*(BTB0_0:BTB3_3)+30*(BTB0_0:BTB3_
    4)+30*(BTB0_0:BTB4_0)+30*(BTB0_0:BTB4_1)+30*(BTB0_0:BTB4_2)+30*(BTB0_0:BTB4_3)+30*(BTB0_0:BT
    B4_4)+30*(BTB0_0:BTB5_0)+30*(BTB0_0:BTB5_1)+30*(BTB0_0:BTB5_2)+30*(BTB0_0:BTB5_3)+30*(BTB0_0
    :BTB5_4)+30*(BTB0_0:BTB6_0)+30*(BTB0_0:BTB6_1)+30*(BTB0_0:BTB6_2)+30*(BTB0_0:BTB6_3)+30*(BTB
    0_0:BTB6_4)+30*(BTB0_0:BTB7_0)+30*(BTB0_0:BTB7_1)+30*(BTB0_0:BTB7_2)+30*(BTB0_0:BTB7_3)+30*(
    BTB0_0:BTB7_4)+30*(BTB0_0:BTB8_0)+30*(BTB0_0:BTB8_1)+30*(BTB0_0:BTB8_2)+30*(BTB0_0:BTB8_3)+3
    0*(BTB0_0:BTB8_4)+30*(BTB0_0:BTB9_0)+30*(BTB0_0:BTB9_1)+30*(BTB0_0:BTB9_2)+30*(BTB0_0:BTB9_3
    )+30*(BTB0_0:BTB9_4)+30*(BTB0_0:BTB10_0)+30*(BTB0_0:BTB10_1)+30*(BTB0_0:BTB10_2)+30*(BTB0_0:
    BTB10_3)+30*(BTB0_0:BTB10_4)+30*(BTB0_0:BTB11_0)+30*(BTB0_0:BTB11_1)+30*(BTB0_0:BTB11_2)+30*
    (BTB0_0:BTB11_3)+30*(BTB0_0:BTB11_4)
10. anchor_residue COM
11.
12. virtual_coordinates_start
13.
14. xyz ROOT +1,+0,+0 +0,+1,+0 0,0,0
15.
16. xyz RA +1,+0,+0 +0,+1,+0 0,0,0
17.
18. xyz RB +1,+0,+0 +0,+1,+0 0,0,0
19.
20. xyz CTA0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-0.85065080835204
    0.0,0.0,0.0
21. xyz CTA1 0.85065080835204,0.0,0.5257311121191335 -
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
22. xyz CTA2 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,0.9510565162951536,-
    0.2628655560595665 0.0,0.0,0.0
23. xyz CTA3 -0.85065080835204,0.0,0.5257311121191335
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
24. xyz CTA4 0.0,0.5257311121191335,-0.85065080835204 -
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
25. xyz CTA5 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-0.2628655560595665,-
    0.16245984811645334 0.0,0.0,0.0
26. xyz CTA6 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-
    0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
27. xyz CTA7 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-
    0.4253254041760199 0.0,0.0,0.0
```

28. xyz CTA8 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0,0.0
29. xyz CTA9 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
30. xyz CTA10 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
31. xyz CTA11 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
32.
33. xyz CTB0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-0.85065080835204 0.0,0.0,0.0,0.0
34. xyz CTB1 0.85065080835204,0.0,0.5257311121191335 -0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0,0.0
35. xyz CTB2 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0,0.0
36. xyz CTB3 -0.85065080835204,0.0,0.5257311121191335 0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0,0.0
37. xyz CTB4 0.0,0.5257311121191335,-0.85065080835204 -0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0,0.0
38. xyz CTB5 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0,0.0
39. xyz CTB6 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0,0.0
40. xyz CTB7 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0,0.0
41. xyz CTB8 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0,0.0
42. xyz CTB9 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
43. xyz CTB10 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
44. xyz CTB11 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
45.
46. xyz PTA0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-0.85065080835204 0.0,0.0,0.0,0.0
47. xyz PTA1 0.85065080835204,0.0,0.5257311121191335 -0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0,0.0
48. xyz PTA2 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0,0.0
49. xyz PTA3 -0.85065080835204,0.0,0.5257311121191335 0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0,0.0
50. xyz PTA4 0.0,0.5257311121191335,-0.85065080835204 -0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0,0.0
51. xyz PTA5 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0,0.0
52. xyz PTA6 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0,0.0
53. xyz PTA7 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0,0.0
54. xyz PTA8 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0,0.0
55. xyz PTA9 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
56. xyz PTA10 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
57. xyz PTA11 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0,0.0
58.
59. xyz PTB0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-0.85065080835204 0.0,0.0,0.0,0.0
60. xyz PTB1 0.85065080835204,0.0,0.5257311121191335 -0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0,0.0
61. xyz PTB2 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0,0.0

62. xyz PTB3 -0.85065080835204,0.0,0.5257311121191335
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
63. xyz PTB4 0.0,0.5257311121191335,-0.85065080835204 -
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
64. xyz PTB5 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-0.2628655560595665,-
    0.16245984811645334 0.0,0.0,0.0
65. xyz PTB6 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-
    0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
66. xyz PTB7 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-
    0.4253254041760199 0.0,0.0,0.0
67. xyz PTB8 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-
    0.5877852522924732 0.0,0.0,0.0
68. xyz PTB9 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-
    0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
69. xyz PTB10 -0.5257311121191335,0.85065080835204,0.0
    0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
70. xyz PTB11 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-
    0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
71.
72. xyz BTA0_0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-
    0.85065080835204 0.0,0.0,0.0
73. xyz BTA0_1 -0.85065080835204,0.0,-0.5257311121191335 0.16245984811645334,-
    0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
74. xyz BTA0_2 -0.85065080835204,0.0,-0.5257311121191335 -0.4253254041760199,-
    0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
75. xyz BTA0_3 -0.85065080835204,0.0,-0.5257311121191335 -
    0.4253254041760199,0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
76. xyz BTA0_4 -0.85065080835204,0.0,-0.5257311121191335
    0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
77. xyz BTA1_0 0.85065080835204,0.0,0.5257311121191335 -
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
78. xyz BTA1_1 0.85065080835204,0.0,0.5257311121191335 -0.5257311121191335,0.0,0.85065080835204
    0.0,0.0,0.0
79. xyz BTA1_2 0.85065080835204,0.0,0.5257311121191335 -0.16245984811645334,-
    0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
80. xyz BTA1_3 0.85065080835204,0.0,0.5257311121191335 0.4253254041760199,-0.5877852522924732,-
    0.6881909602355867 0.0,0.0,0.0
81. xyz BTA1_4 0.85065080835204,0.0,0.5257311121191335 0.4253254041760199,0.5877852522924732,-
    0.6881909602355867 0.0,0.0,0.0
82. xyz BTA2_0 0.85065080835204,0.0,-0.5257311121191335 -
    0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
83. xyz BTA2_1 0.85065080835204,0.0,-0.5257311121191335
    0.4253254041760199,0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
84. xyz BTA2_2 0.85065080835204,0.0,-0.5257311121191335 0.4253254041760199,-
    0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
85. xyz BTA2_3 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,-
    0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
86. xyz BTA2_4 0.85065080835204,0.0,-0.5257311121191335 -0.5257311121191335,0.0,-
    0.85065080835204 0.0,0.0,0.0
87. xyz BTA3_0 -0.85065080835204,0.0,0.5257311121191335
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
88. xyz BTA3_1 -0.85065080835204,0.0,0.5257311121191335 -0.4253254041760199,0.5877852522924732,-
    0.6881909602355867 0.0,0.0,0.0
89. xyz BTA3_2 -0.85065080835204,0.0,0.5257311121191335 -0.4253254041760199,-
    0.5877852522924732,-0.6881909602355867 0.0,0.0,0.0
90. xyz BTA3_3 -0.85065080835204,0.0,0.5257311121191335 0.16245984811645334,-
    0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
91. xyz BTA3_4 -0.85065080835204,0.0,0.5257311121191335 0.5257311121191335,0.0,0.85065080835204
    0.0,0.0,0.0
92. xyz BTA4_0 0.0,0.5257311121191335,-0.85065080835204 -
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
93. xyz BTA4_1 0.0,0.5257311121191335,-0.85065080835204 0.0,0.85065080835204,0.5257311121191335
    0.0,0.0,0.0
94. xyz BTA4_2 0.0,0.5257311121191335,-0.85065080835204
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0

139

95. xyz BTA4_3 0.0,0.5257311121191335,-0.85065080835204 0.5877852522924732,-0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0

96. xyz BTA4_4 0.0,0.5257311121191335,-0.85065080835204 -0.5877852522924732,-0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0

97. xyz BTA5_0 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0

98. xyz BTA5_1 0.0,-0.5257311121191335,0.85065080835204 0.0,-0.85065080835204,-0.5257311121191335 0.0,0.0,0.0

99. xyz BTA5_2 0.0,-0.5257311121191335,0.85065080835204 0.9510565162951536,-0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0

100. xyz BTA5_3 0.0,-0.5257311121191335,0.85065080835204 0.5877852522924732,0.6881909602355867,0.4253254041760199 0.0,0.0,0.0

101. xyz BTA5_4 0.0,-0.5257311121191335,0.85065080835204 -0.5877852522924732,0.6881909602355867,0.4253254041760199 0.0,0.0,0.0

102. xyz BTA6_0 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0

103. xyz BTA6_1 0.0,0.5257311121191335,0.85065080835204 0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0

104. xyz BTA6_2 0.0,0.5257311121191335,0.85065080835204 0.9510565162951536,0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0

105. xyz BTA6_3 0.0,0.5257311121191335,0.85065080835204 0.0,0.85065080835204,-0.5257311121191335 0.0,0.0,0.0

106. xyz BTA6_4 0.0,0.5257311121191335,0.85065080835204 -0.9510565162951536,0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0

107. xyz BTA7_0 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0

108. xyz BTA7_1 0.0,-0.5257311121191335,-0.85065080835204 0.9510565162951536,-0.2628655560595665,0.16245984811645334 0.0,0.0,0.0

109. xyz BTA7_2 0.0,-0.5257311121191335,-0.85065080835204 0.0,-0.85065080835204,0.5257311121191335 0.0,0.0,0.0

110. xyz BTA7_3 0.0,-0.5257311121191335,-0.85065080835204 -0.9510565162951536,-0.2628655560595665,0.16245984811645334 0.0,0.0,0.0

111. xyz BTA7_4 0.0,-0.5257311121191335,-0.85065080835204 -0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0

112. xyz BTA8_0 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0

113. xyz BTA8_1 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0

114. xyz BTA8_2 0.5257311121191335,0.85065080835204,0.0 0.2628655560595665,-0.16245984811645334,0.9510565162951536 0.0,0.0,0.0

115. xyz BTA8_3 0.5257311121191335,0.85065080835204,0.0 0.85065080835204,-0.5257311121191335,0.0 0.0,0.0,0.0

116. xyz BTA8_4 0.5257311121191335,0.85065080835204,0.0 0.2628655560595665,-0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0

117. xyz BTA9_0 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0

118. xyz BTA9_1 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0

119. xyz BTA9_2 0.5257311121191335,-0.85065080835204,0.0 0.2628655560595665,0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0

120. xyz BTA9_3 0.5257311121191335,-0.85065080835204,0.0 0.85065080835204,0.5257311121191335,0.0 0.0,0.0,0.0

121. xyz BTA9_4 0.5257311121191335,-0.85065080835204,0.0 0.2628655560595665,0.16245984811645334,0.9510565162951536 0.0,0.0,0.0

122. xyz BTA10_0 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0

123. xyz BTA10_1 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0

124. xyz BTA10_2 -0.5257311121191335,0.85065080835204,0.0 -0.2628655560595665,-0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0

125. xyz BTA10_3 -0.5257311121191335,0.85065080835204,0.0 -0.85065080835204,-0.5257311121191335,0.0 0.0,0.0,0.0

126. xyz BTA10_4 -0.5257311121191335,0.85065080835204,0.0 -0.2628655560595665,-0.16245984811645334,0.9510565162951536 0.0,0.0,0.0

127. xyz BTA11_0 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-
    0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
128. xyz BTA11_1 -0.5257311121191335,-0.85065080835204,0.0 -
    0.2628655560595665,0.16245984811645334,0.9510565162951536 0.0,0.0,0.0
129. xyz BTA11_2 -0.5257311121191335,-0.85065080835204,0.0 -
    0.85065080835204,0.5257311121191335,0.0 0.0,0.0,0.0
130. xyz BTA11_3 -0.5257311121191335,-0.85065080835204,0.0 -
    0.2628655560595665,0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0
131. xyz BTA11_4 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-
    0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0
132.
133. xyz BTB0_0 -0.85065080835204,0.0,-0.5257311121191335 0.5257311121191335,0.0,-
    0.85065080835204 0.0,0.0,0.0
134. xyz BTB0_1 -0.85065080835204,0.0,-0.5257311121191335 0.16245984811645334,-
    0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
135. xyz BTB0_2 -0.85065080835204,0.0,-0.5257311121191335 -0.4253254041760199,-
    0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
136. xyz BTB0_3 -0.85065080835204,0.0,-0.5257311121191335 -
    0.4253254041760199,0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
137. xyz BTB0_4 -0.85065080835204,0.0,-0.5257311121191335
    0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
138. xyz BTB1_0 0.85065080835204,0.0,0.5257311121191335 -
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
139. xyz BTB1_1 0.85065080835204,0.0,0.5257311121191335 -
    0.5257311121191335,0.0,0.85065080835204 0.0,0.0,0.0
140. xyz BTB1_2 0.85065080835204,0.0,0.5257311121191335 -0.16245984811645334,-
    0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
141. xyz BTB1_3 0.85065080835204,0.0,0.5257311121191335 0.4253254041760199,-
    0.5877852522924732,-0.6881909602355867 0.0,0.0,0.0
142. xyz BTB1_4 0.85065080835204,0.0,0.5257311121191335 0.4253254041760199,0.5877852522924732,-
    0.6881909602355867 0.0,0.0,0.0
143. xyz BTB2_0 0.85065080835204,0.0,-0.5257311121191335 -
    0.16245984811645334,0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
144. xyz BTB2_1 0.85065080835204,0.0,-0.5257311121191335
    0.4253254041760199,0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
145. xyz BTB2_2 0.85065080835204,0.0,-0.5257311121191335 0.4253254041760199,-
    0.5877852522924732,0.6881909602355867 0.0,0.0,0.0
146. xyz BTB2_3 0.85065080835204,0.0,-0.5257311121191335 -0.16245984811645334,-
    0.9510565162951536,-0.2628655560595665 0.0,0.0,0.0
147. xyz BTB2_4 0.85065080835204,0.0,-0.5257311121191335 -0.5257311121191335,0.0,-
    0.85065080835204 0.0,0.0,0.0
148. xyz BTB3_0 -0.85065080835204,0.0,0.5257311121191335
    0.16245984811645334,0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
149. xyz BTB3_1 -0.85065080835204,0.0,0.5257311121191335 -
    0.4253254041760199,0.5877852522924732,-0.6881909602355867 0.0,0.0,0.0
150. xyz BTB3_2 -0.85065080835204,0.0,0.5257311121191335 -0.4253254041760199,-
    0.5877852522924732,-0.6881909602355867 0.0,0.0,0.0
151. xyz BTB3_3 -0.85065080835204,0.0,0.5257311121191335 0.16245984811645334,-
    0.9510565162951536,0.2628655560595665 0.0,0.0,0.0
152. xyz BTB3_4 -0.85065080835204,0.0,0.5257311121191335
    0.5257311121191335,0.0,0.85065080835204 0.0,0.0,0.0
153. xyz BTB4_0 0.0,0.5257311121191335,-0.85065080835204 -
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
154. xyz BTB4_1 0.0,0.5257311121191335,-0.85065080835204
    0.0,0.85065080835204,0.5257311121191335 0.0,0.0,0.0
155. xyz BTB4_2 0.0,0.5257311121191335,-0.85065080835204
    0.9510565162951536,0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
156. xyz BTB4_3 0.0,0.5257311121191335,-0.85065080835204 0.5877852522924732,-
    0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0
157. xyz BTB4_4 0.0,0.5257311121191335,-0.85065080835204 -0.5877852522924732,-
    0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0
158. xyz BTB5_0 0.0,-0.5257311121191335,0.85065080835204 -0.9510565162951536,-
    0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0
159. xyz BTB5_1 0.0,-0.5257311121191335,0.85065080835204 0.0,-0.85065080835204,-
    0.5257311121191335 0.0,0.0,0.0

160. xyz BTB5_2 0.0,-0.5257311121191335,0.85065080835204 0.9510565162951536,-0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0
161. xyz BTB5_3 0.0,-0.5257311121191335,0.85065080835204 0.5877852522924732,0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
162. xyz BTB5_4 0.0,-0.5257311121191335,0.85065080835204 -0.5877852522924732,0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
163. xyz BTB6_0 0.0,0.5257311121191335,0.85065080835204 -0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
164. xyz BTB6_1 0.0,0.5257311121191335,0.85065080835204 0.5877852522924732,-0.6881909602355867,0.4253254041760199 0.0,0.0,0.0
165. xyz BTB6_2 0.0,0.5257311121191335,0.85065080835204 0.9510565162951536,0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0
166. xyz BTB6_3 0.0,0.5257311121191335,0.85065080835204 0.0,0.85065080835204,-0.5257311121191335 0.0,0.0,0.0
167. xyz BTB6_4 0.0,0.5257311121191335,0.85065080835204 -0.9510565162951536,0.2628655560595665,-0.16245984811645334 0.0,0.0,0.0
168. xyz BTB7_0 0.0,-0.5257311121191335,-0.85065080835204 0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0
169. xyz BTB7_1 0.0,-0.5257311121191335,-0.85065080835204 0.9510565162951536,-0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
170. xyz BTB7_2 0.0,-0.5257311121191335,-0.85065080835204 0.0,-0.85065080835204,0.5257311121191335 0.0,0.0,0.0
171. xyz BTB7_3 0.0,-0.5257311121191335,-0.85065080835204 -0.9510565162951536,-0.2628655560595665,0.16245984811645334 0.0,0.0,0.0
172. xyz BTB7_4 0.0,-0.5257311121191335,-0.85065080835204 -0.5877852522924732,0.6881909602355867,-0.4253254041760199 0.0,0.0,0.0
173. xyz BTB8_0 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0
174. xyz BTB8_1 0.5257311121191335,0.85065080835204,0.0 -0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
175. xyz BTB8_2 0.5257311121191335,0.85065080835204,0.0 0.2628655560595665,-0.16245984811645334,0.9510565162951536 0.0,0.0,0.0
176. xyz BTB8_3 0.5257311121191335,0.85065080835204,0.0 0.85065080835204,-0.5257311121191335,0.0 0.0,0.0,0.0
177. xyz BTB8_4 0.5257311121191335,0.85065080835204,0.0 0.2628655560595665,-0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0
178. xyz BTB9_0 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
179. xyz BTB9_1 0.5257311121191335,-0.85065080835204,0.0 -0.6881909602355867,-0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0
180. xyz BTB9_2 0.5257311121191335,-0.85065080835204,0.0 0.2628655560595665,0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0
181. xyz BTB9_3 0.5257311121191335,-0.85065080835204,0.0 0.85065080835204,0.5257311121191335,0.0 0.0,0.0,0.0
182. xyz BTB9_4 0.5257311121191335,-0.85065080835204,0.0 0.2628655560595665,0.16245984811645334,0.9510565162951536 0.0,0.0,0.0
183. xyz BTB10_0 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
184. xyz BTB10_1 -0.5257311121191335,0.85065080835204,0.0 0.6881909602355867,0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0
185. xyz BTB10_2 -0.5257311121191335,0.85065080835204,0.0 -0.2628655560595665,-0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0
186. xyz BTB10_3 -0.5257311121191335,0.85065080835204,0.0 -0.85065080835204,-0.5257311121191335,0.0 0.0,0.0,0.0
187. xyz BTB10_4 -0.5257311121191335,0.85065080835204,0.0 -0.2628655560595665,-0.16245984811645334,0.9510565162951536 0.0,0.0,0.0
188. xyz BTB11_0 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-0.4253254041760199,0.5877852522924732 0.0,0.0,0.0
189. xyz BTB11_1 -0.5257311121191335,-0.85065080835204,0.0 -0.2628655560595665,0.16245984811645334,0.9510565162951536 0.0,0.0,0.0
190. xyz BTB11_2 -0.5257311121191335,-0.85065080835204,0.0 -0.85065080835204,0.5257311121191335,0.0 0.0,0.0,0.0
191. xyz BTB11_3 -0.5257311121191335,-0.85065080835204,0.0 -0.2628655560595665,0.16245984811645334,-0.9510565162951536 0.0,0.0,0.0

```
192.  xyz BTB11_4 -0.5257311121191335,-0.85065080835204,0.0 0.6881909602355867,-
      0.4253254041760199,-0.5877852522924732 0.0,0.0,0.0
193.
194.  virtual_coordinates_stop
195.
196.  connect_virtual JMP0 ROOT RA
197.  connect_virtual JMP1 ROOT RB
198.  connect_virtual JMP2 RA CTA0
199.  connect_virtual JMP3 RA CTA1
200.  connect_virtual JMP4 RA CTA2
201.  connect_virtual JMP5 RA CTA3
202.  connect_virtual JMP6 RA CTA4
203.  connect_virtual JMP7 RA CTA5
204.  connect_virtual JMP8 RA CTA6
205.  connect_virtual JMP9 RA CTA7
206.  connect_virtual JMP10 RA CTA8
207.  connect_virtual JMP11 RA CTA9
208.  connect_virtual JMP12 RA CTA10
209.  connect_virtual JMP13 RA CTA11
210.
211.  connect_virtual JMP14 RB CTB0
212.  connect_virtual JMP15 RB CTB1
213.  connect_virtual JMP16 RB CTB2
214.  connect_virtual JMP17 RB CTB3
215.  connect_virtual JMP18 RB CTB4
216.  connect_virtual JMP19 RB CTB5
217.  connect_virtual JMP20 RB CTB6
218.  connect_virtual JMP21 RB CTB7
219.  connect_virtual JMP22 RB CTB8
220.  connect_virtual JMP23 RB CTB9
221.  connect_virtual JMP24 RB CTB10
222.  connect_virtual JMP25 RB CTB11
223.
224.  connect_virtual JMP26 CTA0 PTA0
225.  connect_virtual JMP27 CTA1 PTA1
226.  connect_virtual JMP28 CTA2 PTA2
227.  connect_virtual JMP29 CTA3 PTA3
228.  connect_virtual JMP30 CTA4 PTA4
229.  connect_virtual JMP31 CTA5 PTA5
230.  connect_virtual JMP32 CTA6 PTA6
231.  connect_virtual JMP33 CTA7 PTA7
232.  connect_virtual JMP34 CTA8 PTA8
233.  connect_virtual JMP35 CTA9 PTA9
234.  connect_virtual JMP36 CTA10 PTA10
235.  connect_virtual JMP37 CTA11 PTA11
236.
237.  connect_virtual JMP38 CTB0 PTB0
238.  connect_virtual JMP39 CTB1 PTB1
239.  connect_virtual JMP40 CTB2 PTB2
240.  connect_virtual JMP41 CTB3 PTB3
241.  connect_virtual JMP42 CTB4 PTB4
242.  connect_virtual JMP43 CTB5 PTB5
243.  connect_virtual JMP44 CTB6 PTB6
244.  connect_virtual JMP45 CTB7 PTB7
245.  connect_virtual JMP46 CTB8 PTB8
246.  connect_virtual JMP47 CTB9 PTB9
247.  connect_virtual JMP48 CTB10 PTB10
248.  connect_virtual JMP49 CTB11 PTB11
249.
250.  connect_virtual JMP50 PTA0 BTA0_0
251.  connect_virtual JMP51 PTA0 BTA0_1
252.  connect_virtual JMP52 PTA0 BTA0_2
253.  connect_virtual JMP53 PTA0 BTA0_3
254.  connect_virtual JMP54 PTA0 BTA0_4
255.  connect_virtual JMP55 PTA1 BTA1_0
```

```
256.   connect_virtual JMP56 PTA1 BTA1_1
257.   connect_virtual JMP57 PTA1 BTA1_2
258.   connect_virtual JMP58 PTA1 BTA1_3
259.   connect_virtual JMP59 PTA1 BTA1_4
260.   connect_virtual JMP60 PTA2 BTA2_0
261.   connect_virtual JMP61 PTA2 BTA2_1
262.   connect_virtual JMP62 PTA2 BTA2_2
263.   connect_virtual JMP63 PTA2 BTA2_3
264.   connect_virtual JMP64 PTA2 BTA2_4
265.   connect_virtual JMP65 PTA3 BTA3_0
266.   connect_virtual JMP66 PTA3 BTA3_1
267.   connect_virtual JMP67 PTA3 BTA3_2
268.   connect_virtual JMP68 PTA3 BTA3_3
269.   connect_virtual JMP69 PTA3 BTA3_4
270.   connect_virtual JMP70 PTA4 BTA4_0
271.   connect_virtual JMP71 PTA4 BTA4_1
272.   connect_virtual JMP72 PTA4 BTA4_2
273.   connect_virtual JMP73 PTA4 BTA4_3
274.   connect_virtual JMP74 PTA4 BTA4_4
275.   connect_virtual JMP75 PTA5 BTA5_0
276.   connect_virtual JMP76 PTA5 BTA5_1
277.   connect_virtual JMP77 PTA5 BTA5_2
278.   connect_virtual JMP78 PTA5 BTA5_3
279.   connect_virtual JMP79 PTA5 BTA5_4
280.   connect_virtual JMP80 PTA6 BTA6_0
281.   connect_virtual JMP81 PTA6 BTA6_1
282.   connect_virtual JMP82 PTA6 BTA6_2
283.   connect_virtual JMP83 PTA6 BTA6_3
284.   connect_virtual JMP84 PTA6 BTA6_4
285.   connect_virtual JMP85 PTA7 BTA7_0
286.   connect_virtual JMP86 PTA7 BTA7_1
287.   connect_virtual JMP87 PTA7 BTA7_2
288.   connect_virtual JMP88 PTA7 BTA7_3
289.   connect_virtual JMP89 PTA7 BTA7_4
290.   connect_virtual JMP90 PTA8 BTA8_0
291.   connect_virtual JMP91 PTA8 BTA8_1
292.   connect_virtual JMP92 PTA8 BTA8_2
293.   connect_virtual JMP93 PTA8 BTA8_3
294.   connect_virtual JMP94 PTA8 BTA8_4
295.   connect_virtual JMP95 PTA9 BTA9_0
296.   connect_virtual JMP96 PTA9 BTA9_1
297.   connect_virtual JMP97 PTA9 BTA9_2
298.   connect_virtual JMP98 PTA9 BTA9_3
299.   connect_virtual JMP99 PTA9 BTA9_4
300.   connect_virtual JMP100 PTA10 BTA10_0
301.   connect_virtual JMP101 PTA10 BTA10_1
302.   connect_virtual JMP102 PTA10 BTA10_2
303.   connect_virtual JMP103 PTA10 BTA10_3
304.   connect_virtual JMP104 PTA10 BTA10_4
305.   connect_virtual JMP105 PTA11 BTA11_0
306.   connect_virtual JMP106 PTA11 BTA11_1
307.   connect_virtual JMP107 PTA11 BTA11_2
308.   connect_virtual JMP108 PTA11 BTA11_3
309.   connect_virtual JMP109 PTA11 BTA11_4
310.
311.   connect_virtual JMP110 PTB0 BTB0_0
312.   connect_virtual JMP111 PTB0 BTB0_1
313.   connect_virtual JMP112 PTB0 BTB0_2
314.   connect_virtual JMP113 PTB0 BTB0_3
315.   connect_virtual JMP114 PTB0 BTB0_4
316.   connect_virtual JMP115 PTB1 BTB1_0
317.   connect_virtual JMP116 PTB1 BTB1_1
318.   connect_virtual JMP117 PTB1 BTB1_2
319.   connect_virtual JMP118 PTB1 BTB1_3
320.   connect_virtual JMP119 PTB1 BTB1_4
```

```
321.  connect_virtual JMP120 PTB2 BTB2_0
322.  connect_virtual JMP121 PTB2 BTB2_1
323.  connect_virtual JMP122 PTB2 BTB2_2
324.  connect_virtual JMP123 PTB2 BTB2_3
325.  connect_virtual JMP124 PTB2 BTB2_4
326.  connect_virtual JMP125 PTB3 BTB3_0
327.  connect_virtual JMP126 PTB3 BTB3_1
328.  connect_virtual JMP127 PTB3 BTB3_2
329.  connect_virtual JMP128 PTB3 BTB3_3
330.  connect_virtual JMP129 PTB3 BTB3_4
331.  connect_virtual JMP130 PTB4 BTB4_0
332.  connect_virtual JMP131 PTB4 BTB4_1
333.  connect_virtual JMP132 PTB4 BTB4_2
334.  connect_virtual JMP133 PTB4 BTB4_3
335.  connect_virtual JMP134 PTB4 BTB4_4
336.  connect_virtual JMP135 PTB5 BTB5_0
337.  connect_virtual JMP136 PTB5 BTB5_1
338.  connect_virtual JMP137 PTB5 BTB5_2
339.  connect_virtual JMP138 PTB5 BTB5_3
340.  connect_virtual JMP139 PTB5 BTB5_4
341.  connect_virtual JMP140 PTB6 BTB6_0
342.  connect_virtual JMP141 PTB6 BTB6_1
343.  connect_virtual JMP142 PTB6 BTB6_2
344.  connect_virtual JMP143 PTB6 BTB6_3
345.  connect_virtual JMP144 PTB6 BTB6_4
346.  connect_virtual JMP145 PTB7 BTB7_0
347.  connect_virtual JMP146 PTB7 BTB7_1
348.  connect_virtual JMP147 PTB7 BTB7_2
349.  connect_virtual JMP148 PTB7 BTB7_3
350.  connect_virtual JMP149 PTB7 BTB7_4
351.  connect_virtual JMP150 PTB8 BTB8_0
352.  connect_virtual JMP151 PTB8 BTB8_1
353.  connect_virtual JMP152 PTB8 BTB8_2
354.  connect_virtual JMP153 PTB8 BTB8_3
355.  connect_virtual JMP154 PTB8 BTB8_4
356.  connect_virtual JMP155 PTB9 BTB9_0
357.  connect_virtual JMP156 PTB9 BTB9_1
358.  connect_virtual JMP157 PTB9 BTB9_2
359.  connect_virtual JMP158 PTB9 BTB9_3
360.  connect_virtual JMP159 PTB9 BTB9_4
361.  connect_virtual JMP160 PTB10 BTB10_0
362.  connect_virtual JMP161 PTB10 BTB10_1
363.  connect_virtual JMP162 PTB10 BTB10_2
364.  connect_virtual JMP163 PTB10 BTB10_3
365.  connect_virtual JMP164 PTB10 BTB10_4
366.  connect_virtual JMP165 PTB11 BTB11_0
367.  connect_virtual JMP166 PTB11 BTB11_1
368.  connect_virtual JMP167 PTB11 BTB11_2
369.  connect_virtual JMP168 PTB11 BTB11_3
370.  connect_virtual JMP169 PTB11 BTB11_4
371.
372.  connect_virtual JMP170 BTA0_0 SUBUNIT B COM
373.  connect_virtual JMP171 BTA0_1 SUBUNIT B COM
374.  connect_virtual JMP172 BTA0_2 SUBUNIT B COM
375.  connect_virtual JMP173 BTA0_3 SUBUNIT B COM
376.  connect_virtual JMP174 BTA0_4 SUBUNIT B COM
377.  connect_virtual JMP175 BTA1_0 SUBUNIT B COM
378.  connect_virtual JMP176 BTA1_1 SUBUNIT B COM
379.  connect_virtual JMP177 BTA1_2 SUBUNIT B COM
380.  connect_virtual JMP178 BTA1_3 SUBUNIT B COM
381.  connect_virtual JMP179 BTA1_4 SUBUNIT B COM
382.  connect_virtual JMP180 BTA2_0 SUBUNIT B COM
383.  connect_virtual JMP181 BTA2_1 SUBUNIT B COM
384.  connect_virtual JMP182 BTA2_2 SUBUNIT B COM
385.  connect_virtual JMP183 BTA2_3 SUBUNIT B COM
```

```
386.    connect_virtual JMP184 BTA2_4 SUBUNIT B COM
387.    connect_virtual JMP185 BTA3_0 SUBUNIT B COM
388.    connect_virtual JMP186 BTA3_1 SUBUNIT B COM
389.    connect_virtual JMP187 BTA3_2 SUBUNIT B COM
390.    connect_virtual JMP188 BTA3_3 SUBUNIT B COM
391.    connect_virtual JMP189 BTA3_4 SUBUNIT B COM
392.    connect_virtual JMP190 BTA4_0 SUBUNIT B COM
393.    connect_virtual JMP191 BTA4_1 SUBUNIT B COM
394.    connect_virtual JMP192 BTA4_2 SUBUNIT B COM
395.    connect_virtual JMP193 BTA4_3 SUBUNIT B COM
396.    connect_virtual JMP194 BTA4_4 SUBUNIT B COM
397.    connect_virtual JMP195 BTA5_0 SUBUNIT B COM
398.    connect_virtual JMP196 BTA5_1 SUBUNIT B COM
399.    connect_virtual JMP197 BTA5_2 SUBUNIT B COM
400.    connect_virtual JMP198 BTA5_3 SUBUNIT B COM
401.    connect_virtual JMP199 BTA5_4 SUBUNIT B COM
402.    connect_virtual JMP200 BTA6_0 SUBUNIT B COM
403.    connect_virtual JMP201 BTA6_1 SUBUNIT B COM
404.    connect_virtual JMP202 BTA6_2 SUBUNIT B COM
405.    connect_virtual JMP203 BTA6_3 SUBUNIT B COM
406.    connect_virtual JMP204 BTA6_4 SUBUNIT B COM
407.    connect_virtual JMP205 BTA7_0 SUBUNIT B COM
408.    connect_virtual JMP206 BTA7_1 SUBUNIT B COM
409.    connect_virtual JMP207 BTA7_2 SUBUNIT B COM
410.    connect_virtual JMP208 BTA7_3 SUBUNIT B COM
411.    connect_virtual JMP209 BTA7_4 SUBUNIT B COM
412.    connect_virtual JMP210 BTA8_0 SUBUNIT B COM
413.    connect_virtual JMP211 BTA8_1 SUBUNIT B COM
414.    connect_virtual JMP212 BTA8_2 SUBUNIT B COM
415.    connect_virtual JMP213 BTA8_3 SUBUNIT B COM
416.    connect_virtual JMP214 BTA8_4 SUBUNIT B COM
417.    connect_virtual JMP215 BTA9_0 SUBUNIT B COM
418.    connect_virtual JMP216 BTA9_1 SUBUNIT B COM
419.    connect_virtual JMP217 BTA9_2 SUBUNIT B COM
420.    connect_virtual JMP218 BTA9_3 SUBUNIT B COM
421.    connect_virtual JMP219 BTA9_4 SUBUNIT B COM
422.    connect_virtual JMP220 BTA10_0 SUBUNIT B COM
423.    connect_virtual JMP221 BTA10_1 SUBUNIT B COM
424.    connect_virtual JMP222 BTA10_2 SUBUNIT B COM
425.    connect_virtual JMP223 BTA10_3 SUBUNIT B COM
426.    connect_virtual JMP224 BTA10_4 SUBUNIT B COM
427.    connect_virtual JMP225 BTA11_0 SUBUNIT B COM
428.    connect_virtual JMP226 BTA11_1 SUBUNIT B COM
429.    connect_virtual JMP227 BTA11_2 SUBUNIT B COM
430.    connect_virtual JMP228 BTA11_3 SUBUNIT B COM
431.    connect_virtual JMP229 BTA11_4 SUBUNIT B COM
432.
433.    connect_virtual JMP230 BTB0_0 SUBUNIT C COM
434.    connect_virtual JMP231 BTB0_1 SUBUNIT C COM
435.    connect_virtual JMP232 BTB0_2 SUBUNIT C COM
436.    connect_virtual JMP233 BTB0_3 SUBUNIT C COM
437.    connect_virtual JMP234 BTB0_4 SUBUNIT C COM
438.    connect_virtual JMP235 BTB1_0 SUBUNIT C COM
439.    connect_virtual JMP236 BTB1_1 SUBUNIT C COM
440.    connect_virtual JMP237 BTB1_2 SUBUNIT C COM
441.    connect_virtual JMP238 BTB1_3 SUBUNIT C COM
442.    connect_virtual JMP239 BTB1_4 SUBUNIT C COM
443.    connect_virtual JMP240 BTB2_0 SUBUNIT C COM
444.    connect_virtual JMP241 BTB2_1 SUBUNIT C COM
445.    connect_virtual JMP242 BTB2_2 SUBUNIT C COM
446.    connect_virtual JMP243 BTB2_3 SUBUNIT C COM
447.    connect_virtual JMP244 BTB2_4 SUBUNIT C COM
448.    connect_virtual JMP245 BTB3_0 SUBUNIT C COM
449.    connect_virtual JMP246 BTB3_1 SUBUNIT C COM
450.    connect_virtual JMP247 BTB3_2 SUBUNIT C COM
```

```
451.  connect_virtual JMP248 BTB3_3 SUBUNIT C COM
452.  connect_virtual JMP249 BTB3_4 SUBUNIT C COM
453.  connect_virtual JMP250 BTB4_0 SUBUNIT C COM
454.  connect_virtual JMP251 BTB4_1 SUBUNIT C COM
455.  connect_virtual JMP252 BTB4_2 SUBUNIT C COM
456.  connect_virtual JMP253 BTB4_3 SUBUNIT C COM
457.  connect_virtual JMP254 BTB4_4 SUBUNIT C COM
458.  connect_virtual JMP255 BTB5_0 SUBUNIT C COM
459.  connect_virtual JMP256 BTB5_1 SUBUNIT C COM
460.  connect_virtual JMP257 BTB5_2 SUBUNIT C COM
461.  connect_virtual JMP258 BTB5_3 SUBUNIT C COM
462.  connect_virtual JMP259 BTB5_4 SUBUNIT C COM
463.  connect_virtual JMP260 BTB6_0 SUBUNIT C COM
464.  connect_virtual JMP261 BTB6_1 SUBUNIT C COM
465.  connect_virtual JMP262 BTB6_2 SUBUNIT C COM
466.  connect_virtual JMP263 BTB6_3 SUBUNIT C COM
467.  connect_virtual JMP264 BTB6_4 SUBUNIT C COM
468.  connect_virtual JMP265 BTB7_0 SUBUNIT C COM
469.  connect_virtual JMP266 BTB7_1 SUBUNIT C COM
470.  connect_virtual JMP267 BTB7_2 SUBUNIT C COM
471.  connect_virtual JMP268 BTB7_3 SUBUNIT C COM
472.  connect_virtual JMP269 BTB7_4 SUBUNIT C COM
473.  connect_virtual JMP270 BTB8_0 SUBUNIT C COM
474.  connect_virtual JMP271 BTB8_1 SUBUNIT C COM
475.  connect_virtual JMP272 BTB8_2 SUBUNIT C COM
476.  connect_virtual JMP273 BTB8_3 SUBUNIT C COM
477.  connect_virtual JMP274 BTB8_4 SUBUNIT C COM
478.  connect_virtual JMP275 BTB9_0 SUBUNIT C COM
479.  connect_virtual JMP276 BTB9_1 SUBUNIT C COM
480.  connect_virtual JMP277 BTB9_2 SUBUNIT C COM
481.  connect_virtual JMP278 BTB9_3 SUBUNIT C COM
482.  connect_virtual JMP279 BTB9_4 SUBUNIT C COM
483.  connect_virtual JMP280 BTB10_0 SUBUNIT C COM
484.  connect_virtual JMP281 BTB10_1 SUBUNIT C COM
485.  connect_virtual JMP282 BTB10_2 SUBUNIT C COM
486.  connect_virtual JMP283 BTB10_3 SUBUNIT C COM
487.  connect_virtual JMP284 BTB10_4 SUBUNIT C COM
488.  connect_virtual JMP285 BTB11_0 SUBUNIT C COM
489.  connect_virtual JMP286 BTB11_1 SUBUNIT C COM
490.  connect_virtual JMP287 BTB11_2 SUBUNIT C COM
491.  connect_virtual JMP288 BTB11_3 SUBUNIT C COM
492.  connect_virtual JMP289 BTB11_4 SUBUNIT C COM
493.
494.
495.  set_dof JMP0 x angle_x
496.
497.
498.  set_dof JMP1 x angle_x
499.
500.  set_jump_group JGRA JMP26 JMP27 JMP28 JMP29 JMP30 JMP31 JMP32 JMP33 JMP34 JMP35 JMP36
      JMP37
501.  set_jump_group JGRB JMP38 JMP39 JMP40 JMP41 JMP42 JMP43 JMP44 JMP45 JMP46 JMP47 JMP48
      JMP49
502.  set_jump_group JGRC JMP170 JMP171 JMP172 JMP173 JMP174 JMP175 JMP176 JMP177 JMP178 JMP179
      JMP180 JMP181 JMP182 JMP183 JMP184 JMP185 JMP186 JMP187 JMP188 JMP189 JMP190 JMP191 JMP192
      JMP193 JMP194 JMP195 JMP196 JMP197 JMP198 JMP199 JMP200 JMP201 JMP202 JMP203 JMP204 JMP205
      JMP206 JMP207 JMP208 JMP209 JMP210 JMP211 JMP212 JMP213 JMP214 JMP215 JMP216 JMP217 JMP218
      JMP219 JMP220 JMP221 JMP222 JMP223 JMP224 JMP225 JMP226 JMP227 JMP228 JMP229
503.  set_jump_group JGRD JMP230 JMP231 JMP232 JMP233 JMP234 JMP235 JMP236 JMP237 JMP238 JMP239
      JMP240 JMP241 JMP242 JMP243 JMP244 JMP245 JMP246 JMP247 JMP248 JMP249 JMP250 JMP251 JMP252
      JMP253 JMP254 JMP255 JMP256 JMP257 JMP258 JMP259 JMP260 JMP261 JMP262 JMP263 JMP264 JMP265
      JMP266 JMP267 JMP268 JMP269 JMP270 JMP271 JMP272 JMP273 JMP274 JMP275 JMP276 JMP277 JMP278
      JMP279 JMP280 JMP281 JMP282 JMP283 JMP284 JMP285 JMP286 JMP287 JMP288 JMP289
```

Creating VLP visualizations in PyMol

Addressing challenges of VLP visualization in PyMol

The VLPs explored in this thesis contain 120 chains and over 100,000 atoms. Both of these

numbers can cause issues when working within a PDB file format, which requires specific

information to be within a specific character windows in an ATOM entry line. PDB format only

allows 5 spaces for atom numbering, so the atoms numbered 100,000+ create line shifts that

make these lines unreadable PyMol, creating strange structural artifacts in some visualization

modes. This is addressed by the following code, which also renames chains to be either B or C

to enable rapid color coding:

```python
1.  #!/usr/bin/env python
2.
3.  import sys
4.
5.  foo = open(sys.argv[1],"rt").readlines()
6.  outfile = open(sys.argv[1][:-4]+"_pymol.pdb","wt")
7.
8.  counter = 200
9.
10. chains = ['B','C']
11. chain_counter = 0
12. res_counter=0
13. previous_resnum = -8
14.
15. for i in range(len(foo)):
16.     if ("ATOM" not in foo[i]) & ("TER" not in foo[i]):
17.         outfile.write(foo[i])
18.         continue
19.     if "ATOM" in foo[i]:
20.         rnum = int(foo[i][22:26])
21.         if rnum != previous_resnum:
22.             previous_resnum = rnum
23.             res_counter+=1
24.         foo[i] = foo[i][:21] + chains[chain_counter] + foo[i][22:]
25.         outfile.write(foo[i])
26.         continue
27.     if "TER" in foo[i]:
28.         chain_counter+=1
29.         chain_counter = chain_counter % len(chains)
30.         if chain_counter==0:
31.             res_counter = 0
32.         outfile.write("TER\n")
33.
34. outfile.close()
35.
```

PDB only allows chain names of a single character. There are 120 chains, which is greater than the number of alphanumeric names available to many programs. For applications which require the ability to select every chain individually (such as the Bio package's PDB parser), it is necessary to assign each chain a unique name, as accomplished by the following script:

```python
1.  #!/usr/bin/env python
2.
3.  import sys
4.
5.  foo = open(sys.argv[1],"rt").readlines()
6.  outfile = open(sys.argv[1][:-4]+"_renumbered.pdb","wt")
7.
8.  counter = 200
9.
10. for i in range(len(foo)):
11.     if ("ATOM" not in foo[i]) & ("TER" not in foo[i]):
12.         outfile.write(foo[i])
13.         continue
14.     if "ATOM" in foo[i]:
15.         foo[i] = foo[i][:21] + chr(counter) + foo[i][22:]
16.         outfile.write(foo[i])
17.         continue
18.     if "TER" in foo[i]:
19.         counter+=1
20.
21. outfile.close()
22.
```

Visualization of viral antigen display

Viral antigen display visualization is useful for assessing antigen spacing around a viral capsid and aiding communication about VLP-based vaccine design. Creating these visualizations, however, is not entirely trivial, as a single antigen copy must be replicated and positioned correctly, based on the positions of N and/or C termini in a given viral capsid structure. We implemented code that efficiently automates this process:

```python
1.  #!/usr/bin/env python
2.
3.  import sys
4.  from Bio import PDB
5.  import numpy as np
```

```
6.  from scipy.spatial.transform import Rotation as R
7.
8.  parser = PDB.PDBParser()
9.  io = PDB.PDBIO()
10. struct = parser.get_structure(sys.argv[1][:-4],sys.argv[1])
11. scaffold = parser.get_structure(sys.argv[2][:-4],sys.argv[2])
12.
13. def get_xyz_of_structure(this_struct):
14.     coords = []
15.
16.     for model in this_struct:
17.         for chain in model:
18.             for residue in chain:
19.                 for atom in residue:
20.                     coords.append(atom.get_coord())
21.
22.     coords = np.array(coords)
23.     return coords
24.
25. def get_xyz_of_chain(this_chain):
26.     coords = []
27.
28.     for residue in this_chain:
29.         for atom in residue:
30.             coords.append(atom.get_coord())
31.
32.     coords = np.array(coords)
33.     return coords
34.
35.
36. def write_new_xyz(this_struct,this_xyz,outfile_name):
37.     counter = 0
38.     for model in this_struct:
39.         for chain in model:
40.             for residue in chain:
41.                 for atom in residue:
42.                     atom.set_coord(this_xyz[counter])
43.                     counter += 1
44.
45.     io.set_structure(this_struct)
46.     io.save(outfile_name)
47.     return
48.
49. def get_residue_xyz(this_struct,resi_num):
50.     # resi_num is number from 1, matching the pdb file
51.     # we have to decrease it by one to match Python's 0-index
52.     # reports xyz of the first atom in the residue
53.     res = [r for r in this_struct.get_residues()]
54.     resi_num -= 1
55.     atoms = [a for a in res[resi_num].get_atoms()]
56.     return atoms[0].get_coord()
57.
58. def get_nterm_xyz(this_struct):
59.     atoms = [a for a in this_struct.get_atoms()]
60.     return atoms[0].get_coord()
61.
62. def get_cterm_xyz(this_struct):
63.     atoms = [a for a in this_struct.get_atoms()]
64.     return atoms[-1].get_coord()
65.
66. def rotation_matrix_from_vectors(vec1, vec2):
67.     """ Find the rotation matrix that aligns vec1 to vec2
68.     :param vec1: A 3d "source" vector
69.     :param vec2: A 3d "destination" vector
70.     :return mat: A transform matrix (3x3) which when applied to vec1, aligns it with vec2.
```

```
71.        """
72.        a, b = (vec1 / np.linalg.norm(vec1)).reshape(3), (vec2 /
   np.linalg.norm(vec2)).reshape(3)
73.        v = np.cross(a, b)
74.        c = np.dot(a, b)
75.        s = np.linalg.norm(v)
76.        kmat = np.array([[0, -v[2], v[1]], [v[2], 0, -v[0]], [-v[1], v[0], 0]])
77.        rotation_matrix = np.eye(3) + kmat + kmat.dot(kmat) * ((1 - c) / (s ** 2))
78.        return rotation_matrix
79.
80. scaffold_xyz = get_xyz_of_structure(scaffold)
81. scaffold_com = np.mean(scaffold_xyz,axis=0)
82.
83. def get_antigen_on_chain_xyz(schain1):
84.        antigen_xyz = get_xyz_of_structure(struct)
85.
86.        # finding a positon that roughly maps the centerline of the antigen
87.        antigen_top_xyz = get_residue_xyz(struct,163)
88.
89.
90.
91.        antigen_cterm_xyz = get_cterm_xyz(struct)
92.        scaffold_nterm_xyz = get_nterm_xyz(schain1)
93.
94.        source_antigen_vector = antigen_top_xyz - antigen_cterm_xyz
95.        destination_scaffold_vector = scaffold_nterm_xyz - scaffold_com
96.
97.        #print(source_antigen_vector)
98.        #print(destination_scaffold_vector)
99.        #quit()
100.
101.        rot_matrix = rotation_matrix_from_vectors(source_antigen_vector,
102.                                                  destination_scaffold_vector)
103.
104.        r = R.from_matrix(rot_matrix)
105.        #if(antigen_cterm_xyz.dot(scaffold_nterm_xyz)
106.        #        /(np.linalg.norm(antigen_cterm_xyz)*np.linalg.norm(scaffold_nterm_xyz)) <
   0.7):
107.        antigen_xyz = r.apply(antigen_xyz)
108.        #antigen_xyz = np.transpose(rot_matrix.dot(np.transpose(antigen_xyz)))
109.
110.        antigen_cterm_xyz = antigen_xyz[-1] # note: struct is not yet updated with
111.                                            # new coords to the method doesn't work here
112.        antigen_xyz += (scaffold_nterm_xyz-antigen_cterm_xyz)
113.        return antigen_xyz
114.
115.
116.
117.   def get_updated_structure(this_struct,this_xyz,this_id_counter):
118.        counter = 0
119.        this_struct = this_struct.copy()
120.        for model in this_struct:
121.            for chain in model:
122.                for residue in chain:
123.                    for atom in residue:
124.                        atom.set_coord(this_xyz[counter])
125.                        counter += 1
126.
127.        this_struct.id = chr(this_id_counter)
128.        this_id_counter += 1
129.        this_struct.child_dict[0].serial_num=0
130.        this_struct.child_dict[0].child_dict[' '].id=chr(this_id_counter)
131.        return this_struct
132.
133.   id_counter = 200
```

```python
134.   chains = [c for c in scaffold.get_chains()]
135.
136.   antigens = []
137.   for i in 2*np.arange(60)+1: # update this depending on with C-term you want
138.       print(chains)
139.       print(i)
140.       antigens.append(get_antigen_on_chain_xyz(chains[i]))
141.
142.   for i, a in enumerate(antigens):
143.       antigens[i] = get_updated_structure(struct,a,id_counter)
144.       id_counter += 1
145.
146.   for a in antigens[1:]:
147.       for model in a:
148.           for chain in model:
149.               antigens[0].child_dict[0].add(chain)
150.
151.   io.set_structure(antigens[0])
152.   outfile_name = "i32_B_cterm_rbd.pdb"
153.
154.   io.save(outfile_name)
155.
156.   # renumber output file because pymol is dumb
157.   foo = open(outfile_name,"rt").readlines()
158.   outfile = open(outfile_name,"wt")
159.
160.   for i in range(len(foo)):
161.       if ("ATOM" not in foo[i]) & ("TER" not in foo[i]):
162.           outfile.write(foo[i])
163.           continue
164.       if "ATOM" in foo[i]:
165.           atom_nr = int(foo[i].split()[1])
166.           if atom_nr >= 100000:
167.               foo[i] = foo[i][:9]+foo[i][10:] # removing a zero since this
168.                                               # causes dumb pymol issues
169.           foo[i] = foo[i][:21] + "A" + foo[i][22:]
170.           outfile.write(foo[i])
171.           continue
172.       if "TER" in foo[i]:
173.           outfile.write("TER\n")
174.
175.   outfile.close()
176.
```

Compute environment setup on AWS

ParallelCluster config file

As detailed in Chapter 5, we found that AWS ParallelCluster enabled the creation of an HPC-like

compute environment when properly configured. The configuration details necessary to

achieve this setup are detailed in the config file below:

```
1.  [aws]
2.  aws_region_name = us-east-2
3.
4.  [aliases]
5.  ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
6.
7.  [global]
8.  cluster_template = default
9.  update_check = true
10. sanity_check = true
11.
12. [cluster default]
13. key_name = aws1
14. scheduler = slurm
15. master_instance_type = t2.micro
16. base_os = alinux2
17. vpc_settings = default
18. queue_settings = compute
19. fsx_settings = fsx
20.
21. [vpc default]
22. vpc_id = vpc-a1cc5fca
23. master_subnet_id = subnet-49c37822
24. vpc_security_group_id = sg-cd1e66ba
25.
26. [queue compute]
27. enable_efa = false
28. enable_efa_gdr = false
29. compute_resource_settings = default
30.
31. [compute_resource default]
32. instance_type = t2.2xlarge
33. max_count = 1000
34.
35. [fsx fsx]
36. shared_dir = /fsx
37. fsx_fs_id = fs-0fe463815adbfc334
38.
```

Further setup instructions can be found here: https://jiaweizhuang.github.io/blog/aws-hpc-

guide/

Loading FSX during the launch of a new instance

During the launch of a new AWS EC2 instance, special instructions must be entered into the "User data" field during instance configuration to ensure automatic and proper loading of the persistent file system. The necessary lines are included below:

```
1.  #cloud-boothook
2.  #!/bin/bash
3.  fsx_dnsname=fs-0fe463815adbfc334.fsx.us-east-2.amazonaws.com
4.  fsx_mountname=4eykjbmv
5.  fsx_mountpoint=/fsx
6.  sudo amazon-linux-extras install -y lustre2.10
7.  sudo mkdir -p "$fsx_mountpoint"
8.  sudo mount -t lustre ${fsx_dnsname}@tcp:/${fsx_mountname} ${fsx_mountpoint} -o noatime,flock
9.
```

It is important to note that the "#cloud-boothook" line was not included in the AWS documentation but appears necessary to achieve automatic loading during the launch of a new instance.

Slurm file header for AWS

Our AWS parallel cluster is configured to utilize a slurm scheduler, common to many HPC applications. Still, without the correct slurm file header, computational resource allocation appears to react dynamically to the perceived computational requirements of all jobs on the cluster. This can create memory or computational shortages during rapid spikes in computational demand, resulting in job failures. The "--exclusive" flag must be added to the header of each slurm file to ensure the availability of the desired computational resources for each job as shown below:

```
1.  #!/bin/bash
2.  #SBATCH -t 3-00:00
3.  #SBATCH -N 1
4.  #SBATCH --exclusive
5.
```

Configuring an AWS security group and Schrodinger license server

AWS security groups govern accessibility rules for AWS application elements, both within AWS

and between AWS and outside network traffic. We configured a default security group,

common to all relevant AWS elements (FSX, Parallel Cluster, new EC2 instances, etc.), to ensure

seamless communication between elements. In addition to the default rules, we added TCP

protocols to enable ssh access on port 22 from any external IP address and opened TCP traffic

on ports 27008 and 53000 to enable communication between our Schrodinger license server

and desktop clients. It is worth noting that opening these Schrodinger ports ensures that

communication to and from AWS cannot be blocked by AWS; it does not ensure that TCP

communication cannot be blocked by a local network with a stringent firewall. As a result, it is

possible for license communication to work at home or on mobile hotspot but not when

connected to the MIT wireless network, for example. We should also note that the Schrodinger

license server can only be installed on one physical node, so losing access to a specific compute

node could necessitate acquisition and installation of a fresh license from Schrodinger. For this

reason, a termination protection was added to the Schrodinger license server node.

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☐	–	sgr-0660cf7a91e425adb	IPv4	Custom TCP	TCP	27008	0.0.0.0/0
☐	–	sgr-08a2f9eb042dec537	IPv4	SSH	TCP	22	0.0.0.0/0
☐	–	sgr-0a6794749d7d08...	–	All traffic	All	All	sg-cd1e66ba
☐	–	sgr-0031b8b53a292a...	IPv4	Custom TCP	TCP	53000	0.0.0.0/0

Inbound rules (4) · Manage tags · Edit inbound rules

Figure A.6 AWS inbound security group rules

Figure A.7 AWS outbound security group rules