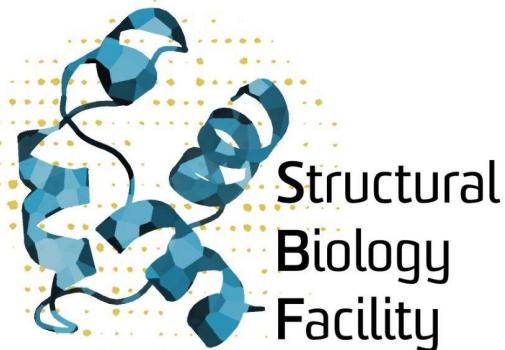




UNSW  
SYDNEY

# Serving next generation Structural Biology with GPUs



**SBF Team (L-R) :** Nathan Glades, Keiran Rowell, Kate Michie, Josh Caley

CRICOS Provider Code 00098G



UNSW  
SYDNEY

# The revolution in Computational Structural Biology

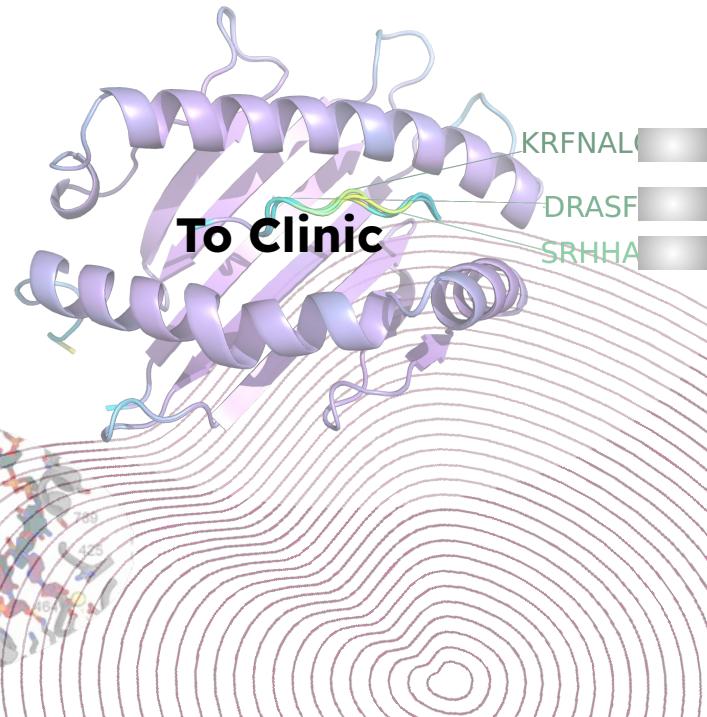
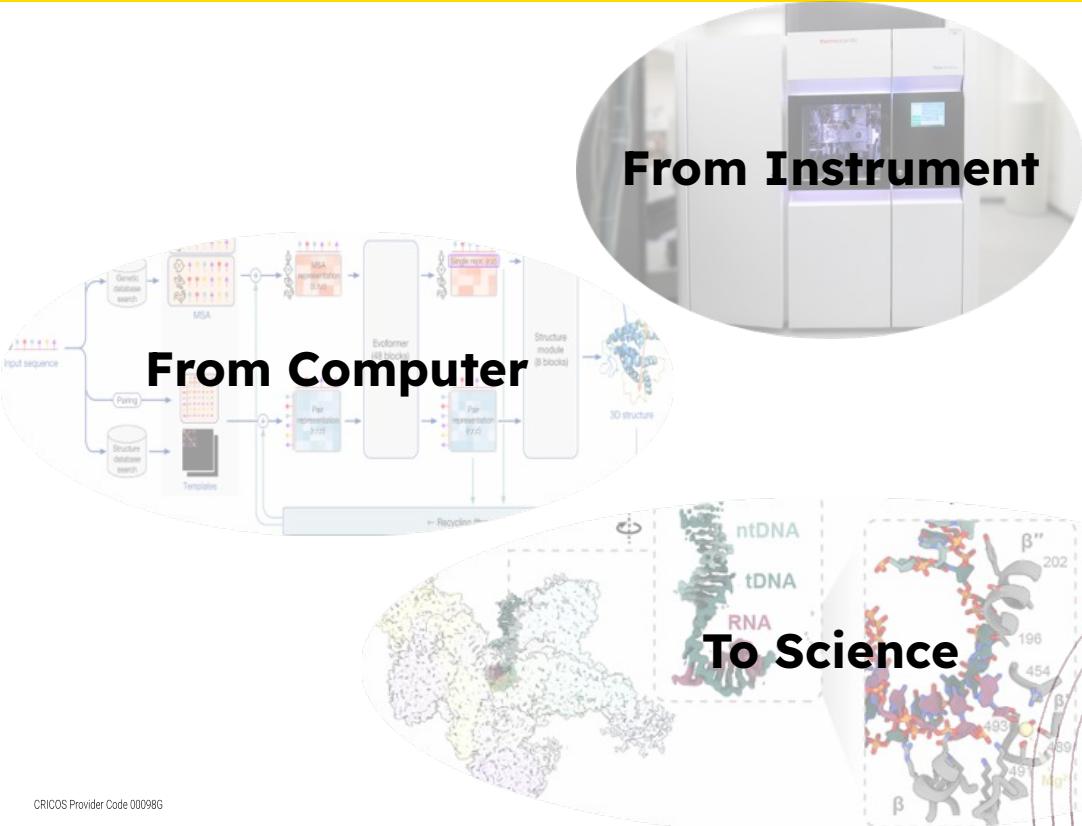
THE NOBEL PRIZE  
IN CHEMISTRY 2024



David  
Baker

Demis  
Hassabis

John M.  
Jumper



# Deep Learning Biomolecular structures – Uses

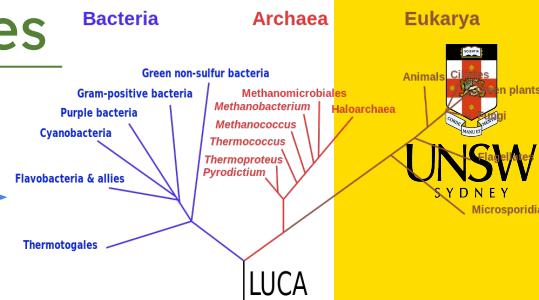
Archaeal evolution - Michie/Burns Labs - ESMFold + FoldSeek



Sequence whole genome  
4622 proteins



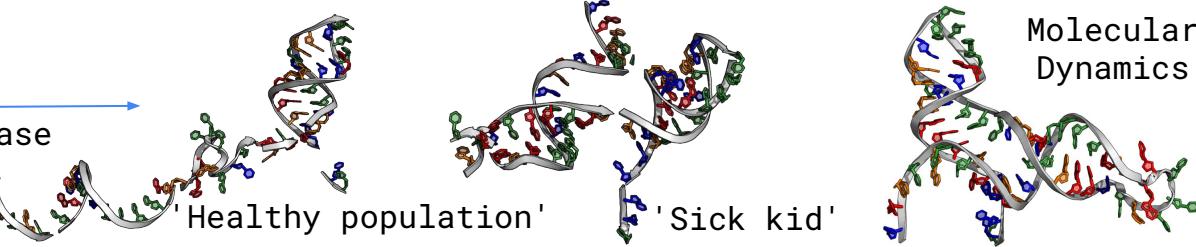
# *Structural* AFDB-search



Medical genomics – Oates Lab – RFoldNA

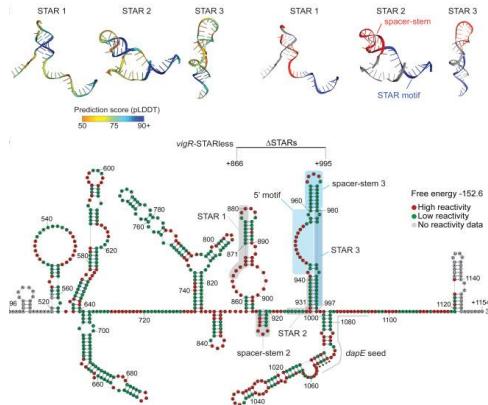


172 variants  
Neurological disease  
(genetic) 



# Molecular Dynamics

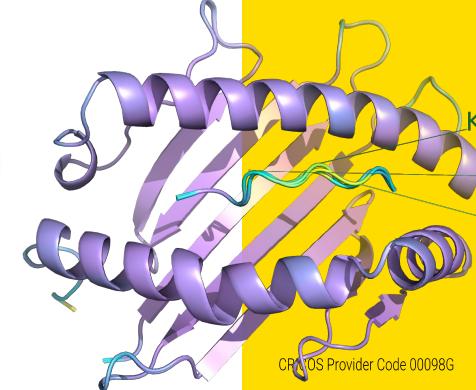
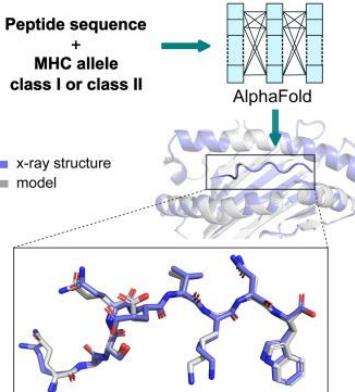
Antibiotic Resistance – Tree Lab – RFoldNA



'Golden Staph'  
vancomycin  
tolerance via  
RNA structures

**Immune response – Tedla Lab – T(Cell)Fold**

**Peptide sequence  
+  
MHC allele  
class I or class II**



# Building the Australian Computational Structural Biology Community

[australian-structural-biology-computing.github.io/website/](https://australian-structural-biology-computing.github.io/website/)

Australian Structural Biology Computing

Home GitHub Search StructBio Computing

Guide AlphaFold2 How-to Guide ↗ ! ↲

AlphaFold2 (AF2) has undoubtedly revolutionised the world of structural biology, allowing for the rapid prediction of proteins and protein complexes. In this guide I aim to equip you to understand how the algorithm works and how to interpret the output data.

To do this there are 3 main sections:

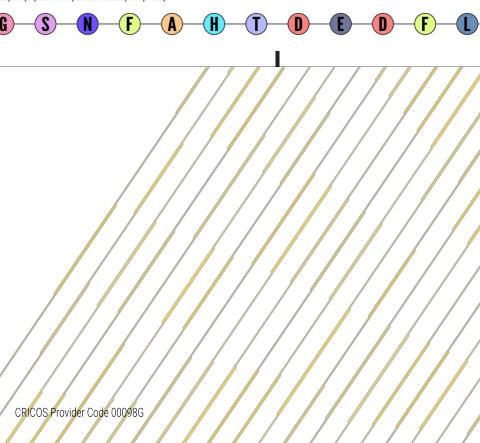
1. AlphaFold2 - the basics
2. Running a Structure Prediction, and
3. Interpreting the Output Files

This guide is designed as a supplement to EMBL's excellent course on AF2.

### AlphaFold2 - the basics

To start to get to grips with how AF2 goes about generating a protein model, it is helpful to understand Anfinsen's dogma and Levinthal's paradox.

Anfinsen's dogma: "at least for a small globular protein in its standard physiological environment, the native structure is determined only by the protein's amino acid sequence". This means given a random amino acid sequence it should be possible to predict the secondary structure it adopts (alpha helix, beta strand, loop etc).



What AlphaFold data should I include?

On this page

- AlphaFold2 - the basics
- Running a structure prediction
- Interpreting the output files

Best practices for presenting and sharing AlphaFold models in a paper ↗ ! ↲

AlphaFold structural models are appearing in papers more frequently. As such, it's important that the scientific community agree on:

- General guidelines on how to best present AlphaFold models: AlphaFold models presented without sufficient information might mislead readers into thinking the model confidence is higher than it really is.
- Guidelines for sharing AlphaFold output files: AlphaFold models are computationally expensive to generate, and some readers might not have the resources to re-generate the model.

For these reasons, we have distilled our knowledge, and that of the broader field, into a guide for presenting and sharing AlphaFold models in papers. The guide is not intended to be exhaustive or dogmatically rigid, and we expect it will evolve over time. Our goals with this guide are to:

- Help newcomers get up to speed with how-to present AlphaFold models, and
- Encourage wider discussion among the structural biology community.

Please reach out to us if you have suggestions.

CRICOS Provider Code 00098G

Contributors

The projects included here represent a joint effort by the following people at multiple Australian institutions.



James Lingford  
Research Officer, Greening Lab, Monash University

[Email](#) [GitHub](#) [LinkedIn](#)



Johan Gustafsson  
Engagement, Australian BioCommons, University of Melbourne

[Email](#) [GitHub](#) [LinkedIn](#)



Kate Michie  
Chief Scientist, Structural Biology Facility UNSW

[Email](#) [GitHub](#) [LinkedIn](#)



Keiran Rowell  
Computational Scientist, Structural Biology Facility UNSW

[Email](#) [GitHub](#) [LinkedIn](#)



Michael Healy  
Postdoctoral Research Fellow, University of Queensland

[Email](#) [GitHub](#) [LinkedIn](#)

Australian Structural Biology Computing

Home GitHub Search StructBio Computing

On this page

- What AlphaFold data should I include?
- Best practices for presenting models
- Other good practices
- What should I include in the methods section?
- What AlphaFold data should I share?



UNSW SYDNEY



# Problems Facing Computational Protein Folding Adoption

## Command line

```
-zsh
z3545907@L-M222T4CK6X ~ % echo 'Hello, World!' | cowsay
< Hello, World! >
-----
 \ ^__^
  (oo)\----_
   (__)\       )\/\
    ||----w |
     ||     |
z3545907@L-M222T4CK6X ~ %
```

Command line can be very finicky.  
Researchers shouldn't have to touch it.

## File Formats

- ─ samplesheet.csv
- {} samplesheet.json
- ≡ samplesheet.tsv
- ! samplesheet.yml

Different folding classes want different file formats and different parameters.

## Resource Allocation



What resources should you allocate to your job? How much do you need?

# Solutions: Command Line



UNSW  
SYDNEY

The screenshot shows a web browser window titled "Proteinifold - Open OnDemand" at the URL [https://kod.restech.unsw.edu.au/pun/sys/dashboard/batch\\_connect/](https://kod.restech.unsw.edu.au/pun/sys/dashboard/batch_connect/). The page displays the "Interactive Apps" section of the Katana OnDemand interface. The "Proteinifold" application is selected. The main content area shows the "Proteinifold version: 31abf0c" and a "Samplesheet" input field containing the sequence "MNIFEMLRIDEGLRLKIYKDTEGYYTIGIHLTKSPSLNA". Below it, under "Acceptable inputs", there are two bullet points: "Directory containing Fasta files /srv/scratch/z3545907/Experiment" and "Amino acid sequence MPGAFSQNSSKRRAVLPRSHR". The "Run Name" field contains "Test Custom Samplesheet ESMFold", and the "Method" dropdown is set to "ESMFold", with two additional options: "AlphaFold 2.3 High Accuracy, Slower" and "ColabFold Medium Accuracy, Faster". The left sidebar has sections for "Interactive Apps [Sandbox]" (Desktops, FastX Desktop, JupyterLab, RStudio Server), "Biology", and "Proteinifold". The top navigation bar includes links for Bitwarden Web Vault, MyNCI, CryoSPARC, Nextflow, Alphafold, Pawsey, Calendar - Nathan, and Confluence.

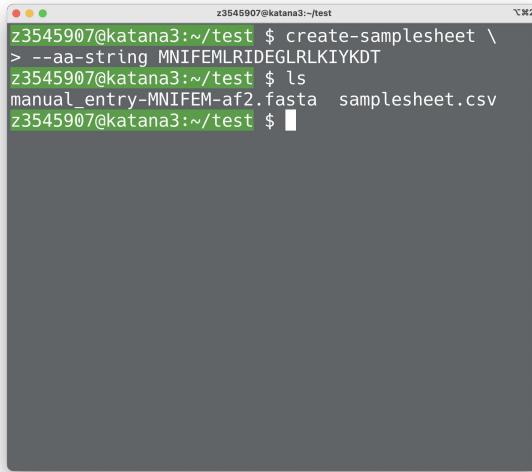
## OnDemand

- Provides a nice frontend for users
- Easily manage previous and currently running jobs
- Integrates with local cluster schedulers
- Easily add, remove and modify fields

# Solutions: Samplesheet



UNSW  
SYDNEY

A screenshot of a terminal window titled "z3545907@katana3:~/test". The window shows the following command-line session:

```
z3545907@katana3:~/test $ create-samplesheet \
> --aa-string MNIFEMLRIDEGLRLKIYKDT
z3545907@katana3:~/test $ ls
manual_entry-MNIFEM-af2.fasta  samplesheet.csv
z3545907@katana3:~/test $
```

The terminal has a dark grey background and white text.

## **samplesheet-utils**

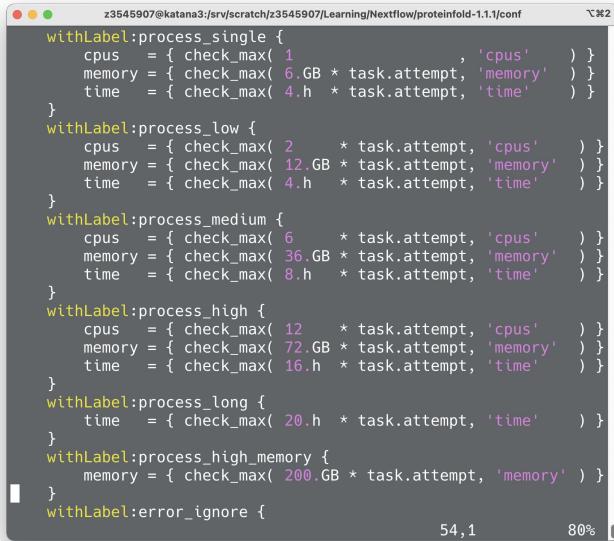
- Easily create samplesheets from directories containing FASTA files
- Create a samplesheet from manual input sequence
- Translate samplesheets to different formats
- Easily integratable into Nextflow pipelines

<https://github.com/Australian-Structural-Biology-Computing/create-samplesheet>

# Solutions: Resource Allocation



UNSW  
SYDNEY



A screenshot of a terminal window titled "z3545907@katana3:/srv/scratch/z3545907/Learning/Nextflow/proteinifold-1.1.1/conf". The window displays a Nextflow configuration script. The code defines several resource allocation strategies using the `withLabel` keyword:

```
withLabel:process_single {
    cpus = { check_max( 1           , 'cpus' ) }
    memory = { check_max( 6.GB * task.attempt, 'memory' ) }
    time = { check_max( 4.h * task.attempt, 'time' ) }
}
withLabel:process_low {
    cpus = { check_max( 2       * task.attempt, 'cpus' ) }
    memory = { check_max( 12.GB * task.attempt, 'memory' ) }
    time = { check_max( 4.h * task.attempt, 'time' ) }
}
withLabel:process_medium {
    cpus = { check_max( 6       * task.attempt, 'cpus' ) }
    memory = { check_max( 36.GB * task.attempt, 'memory' ) }
    time = { check_max( 8.h * task.attempt, 'time' ) }
}
withLabel:process_high {
    cpus = { check_max( 12      * task.attempt, 'cpus' ) }
    memory = { check_max( 72.GB * task.attempt, 'memory' ) }
    time = { check_max( 16.h * task.attempt, 'time' ) }
}
withLabel:process_long {
    time = { check_max( 20.h * task.attempt, 'time' ) }
}
withLabel:process_high_memory {
    memory = { check_max( 200.GB * task.attempt, 'memory' ) }
}
withLabel:error_ignore {
```

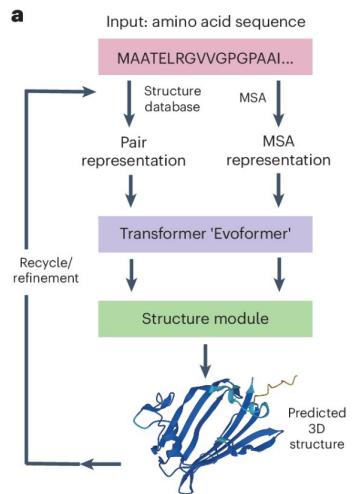
## Nextflow

- Can dynamically choose how many resources to allocate depending on the task
- Automatically parallelise jobs
- Automatically retry with more resources if jobs run out of memory, etc.

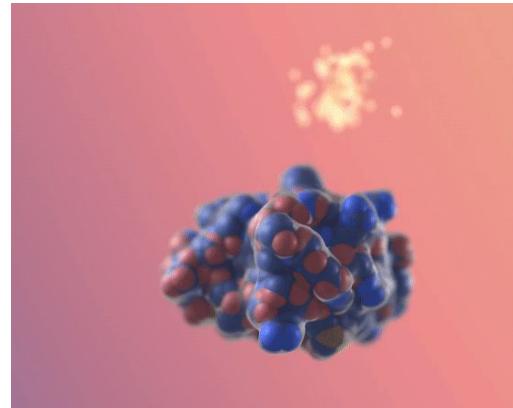


# Advancements in Structural Software: Predicting, Generating, and Aligning Biomolecular Structures

## Structure Prediction



## Diffusion Models



## Structure Alignment



# Future Directions in Binder Design and Structural Biology

