The Fatgraph Models of Proteins and Their Applications in The Protein Folding Problem

Yuki Koyanagi

PhD Dissertation

Supervisors: Jørgen Ellegaard Andersen
Gergely Bérczi

Department of Mathematics
Aarhus University
January, 2021

Corrections

Below follow corrections found after submitting this disertation. The page numbers refers to the pages in this PDF.

- Page 8. The matrix model should read;

$$Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r}_{\boldsymbol{i},\boldsymbol{j};p}) = \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3} dAdBdM\ \mathrm{e}^{-N\mathrm{Tr}V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)}.$$

- Page 8. The potential should read;

$$\begin{aligned}
&V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)\\
&= AB+\frac{1}{2}M^2-\sum_{i\geq 0}y^i\eta\Big\{\\
&\quad+\alpha_i^{(1)}(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^i(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^i\\
&\quad+\alpha_i^{(2)}(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^i(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^i\\
&\quad+\beta_i^{(1)}(M+\eta^{-1/2}\Lambda)\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&\qquad\times(A+y^{-1/2}\Lambda_1)^{i\%2}(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^{i\%2}\\
&\qquad\times\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&\quad+\beta_i^{(2)}(M+\eta^{-1/2}\Lambda)\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\\
&\qquad\times(B+y^{-1/2}\Lambda_2)^{i\%2}(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\qquad\times\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\Big\}
\end{aligned}$$

- Page 10, 1st line should read; "we can improve the prediction accuracy of an existing software programme (table 3)."
- Page 11, 6th line from the bottom. "to 1% of the volum of SO(3) (table 5)".
- Page 25, 18th line from the top. "...the random matrix under consideration, is well-defined."
- Page 54, 10th line from the top. "The total number l of unpaired".
- Page 54, 12th line from the top.

$$\frac{l}{2}=\sum_{K\geq 1}\sum_{(i_1,\ldots,i_K)}\sum_{(j_1,\ldots,j_K)}\sum_{L=1}^{K}i_L\ell_{(i_1,\ldots,i_K)(j_1,\ldots,j_K)}$$

- Page 55, equation 42.

$$\langle A_{ab}B_{cd}\rangle=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^2)}\int_{\mathcal{H}_N^2}dAdB\ A_{ab}B_{cd}\mathrm{e}^{-\mathrm{Tr}AB}=\delta_{ad}\delta_{bc},$$

- Page 55–57, the integral is over $\mathcal{H}_N^2$, not $\mathcal{H}_N^{\otimes 2}$.
- Page 60, definition 5. The partition function should read;

$$
\begin{aligned}
&Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3} dAdBdM\,\exp\Bigg[-N\mathrm{Tr}\Bigg(AB+\frac{1}{2}M^2\\
&\quad+\sum_{i\geq 0}y^i\Bigg\{\alpha_i^{(1)}M(A+y^{-1/2}\Lambda_1)^iM(B+y^{-1/2}\Lambda_2)^i\\
&\qquad\qquad+\alpha_i^{(2)}M(B+y^{-1/2}\Lambda_2)^iM(A+y^{-1/2}\Lambda_1)^i\\
&\qquad\qquad+\beta_i^{(1)}M\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\qquad\qquad\quad\times M(B+y^{-1/2}\Lambda_2)^{i\%2}\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&\qquad\qquad+\beta_i^{(2)}M\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}(B+y^{-1/2}\Lambda_2)^{i\%2}\\
&\qquad\qquad\quad\times M(A+y^{-1/2}\Lambda_1)^{i\%2}\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\Bigg\}\Bigg)\Bigg]\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3} dAdBdM\;\mathrm{e}^{-N\mathrm{Tr}V_{\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2)},
\end{aligned}
$$

  where $[i/2]$ denotes the integer part of $i/2$, and $i\%2$ denotes i modulo 2.
- The proof for theorem 4.8 should be changed accordingly.

  Proof. First, we consider the derivative $\partial/\partial\Lambda_1$ of the partition function $Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})$.

$$
\begin{aligned}
&\frac{\partial}{\partial\Lambda_{1ba}}Z_N(y;\boldsymbol{\alpha}_i^{(1)},\boldsymbol{\alpha}_i^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3} dAdBdM\;N\sum_{i\geq 1}y^{i-1/2}\\
&\quad\times\Bigg(\alpha_i^{(1)}\sum_{k=0}^{i-1}\big((A+y^{-1/2}\Lambda_1)^kM(B+y^{-1/2}\Lambda_2)^iM(A+y^{-1/2}\Lambda_1)^{i-k-1}\big)_{ab}\\
&\qquad+\alpha_i^{(2)}\sum_{k=0}^{i-1}\big((A+y^{-1/2}\Lambda_1)^kM(B+y^{-1/2}\Lambda_2)^iM(A+y^{-1/2}\Lambda_1)^{i-k-1}\big)_{ab}\\
&\qquad+\beta_i^{(1)}\Big\{\sum_{k=0}^{[i/2]-1}(B+y^{-1/2}\Lambda_2)((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^k\\
&\qquad\qquad\times(A+y^{-1/2}\Lambda_1)^{i\%2}M(B+y^{-1/2}\Lambda_2)^{i\%2}((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]}\\
&\qquad\qquad\times M((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]-k-1}\\
&\qquad\quad+M(B+y^{-1/2}\Lambda_2)((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]}\\
&\qquad\qquad\times M((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]}(i\%2)\\
&\qquad\quad+\sum_{k=0}^{[i/2]-1}(B+y^{-1/2}\Lambda_2)((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^k
\end{aligned}
$$

$$
\begin{aligned}
&\times M((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]}(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\times M(B+y^{-1/2}\Lambda_2)^{i\%2}((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{[i/2]-k-1}\Big\}_{ab}\\
&+\beta_i^{(2)}\Big\{\sum_{k=0}^{[i/2]-1}((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^k(B+y^{-1/2}\Lambda_2)^{i\%2}\\
&\times M(A+y^{-1/2}\Lambda_1)^{i\%2}((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]}\\
&\times M((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]-k-1}(B+y^{-1/2}\Lambda_2)\\
&+((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]}M\\
&\times((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]}(B+y^{-1/2}\Lambda_2)M(i\%2)\\
&+\sum_{k=0}^{[i/2]-1}((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^kM((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]}\\
&\times(B+y^{-1/2}\Lambda_2)^{i\%2}M(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\times((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{[i/2]-k-1}(B+y^{-1/2}\Lambda_2)\Big\}_{ab}\Big)\\
&\times \mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2)}\\
=&\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3}dXdYdM\ N\sum_{i\geq 1}y^{i-1/2}\\
&\times\Big(\alpha_i^{(1)}\sum_{k=0}^{i-1}X^kMY^iMX^{i-k-1}\\
&+\alpha_i^{(2)}\sum_{k=0}^{i-1}X^kMY^iMX^{i-k-1}\\
&+\beta_i^{(1)}\sum_{k=0}^{[i/2]-1}\big(Y(XY)^kX^{i\%2}MY^{i\%2}(XY)^{[i/2]}M(XY)^{[i/2]-k-1}\\
&+MY(XY)^{[i/2]}M(XY)^{[i/2]}(i\%2)\\
&+Y(XY)^kM(XY)^{[i/2]}X^{i\%2}MY^{i\%2}(XY)^{[i/2]-k-1}\big)\\
&+\beta_i^{(2)}\sum_{k=0}^{[i/2]-1}\big((YX)^kY^{i\%2}MX^{i\%2}(YX)^{[i/2]}M(YX)^{[i/2]-k-1}Y\\
&+(YX)^{[i/2]}M(YX)^{[i/2]}YM(i\%2)\\
&+(YX)^kM(YX)^{[i/2]}Y^{i\%2}MX^{i\%2}(YX)^{[i/2]-k-1}Y\big)\Big)_{ab}\\
&\times\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)},
\end{aligned}
$$

where $X=A+y^{-1/2}\Lambda_1$, $Y=B+y^{-1/2}\Lambda_2$, and

$$
\begin{aligned}
&W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)\\
&=(X-y^{-1/2}\Lambda_1)(Y-y^{-1/2}\Lambda_2)+\frac{1}{2}M^2\\
&\quad-\sum_{i\geq 0}y^i\{\alpha_i^{(1)}MX^iMY^i+\alpha_i^{(2)}MY^iMX^i
\end{aligned}
$$

$$
\begin{aligned}
&+\beta_i^{(1)} M(XY)^{[i/2]} X^{i\%2} M Y^{i\%2} (XY)^{[i/2]} \\
&+\beta_i^{(2)} M(YX)^{[i/2]} Y^{i\%2} M X^{i\%2} (YX)^{[i/2]}\}.
\end{aligned}
$$

We now compute the derivative $\sum_{a,b=1}^{N} \partial/\partial\Lambda_{2ab}$ to find

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2} Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}_i^{(1)},\boldsymbol{\beta}_i^{(2)},\boldsymbol{r_{ij}}) \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)} \int_{\mathcal{H}_N^3} dXdYdM\ N^2 \sum_{i\geq 1} y^{i-1} \\
&\quad \times \mathrm{Tr}\Bigg(\Big(\alpha_i^{(1)} \sum_{k=0}^{i-1} X^k M Y^i M X^{i-k-1} + \alpha_i^{(2)} \sum_{k=0}^{i-1} X^k M Y^i M X^{i-k-1} \\
&\qquad + \beta_i^{(1)} \sum_{k=0}^{[i/2]-1} \big(Y(XY)^k X^{i\%2} M Y^{i\%2} (XY)^{[i/2]} M (XY)^{[i/2]-k-1} \\
&\qquad\quad + MY(XY)^{[i/2]} M (XY)^{[i/2]} (i\%2) \\
&\qquad\quad + Y(XY)^k M (XY)^{[i/2]} X^{i\%2} M Y^{i\%2} (XY)^{[i/2]-k-1} \big) \\
&\qquad + \beta_i^{(2)} \sum_{k=0}^{[i/2]-1} \big((YX)^k Y^{i\%2} M X^{i\%2} (YX)^{[i/2]} M (YX)^{[i/2]-k-1} Y \\
&\qquad\quad + (YX)^{[i/2]} M (YX)^{[i/2]} Y M (i\%2) \\
&\qquad\quad + (YX)^k M (YX)^{[i/2]} Y^{i\%2} M X^{i\%2} (YX)^{[i/2]-k-1} Y \big) \\
&\qquad \times (X - y^{-1/2}\Lambda_1) \Bigg) \mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)} \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)} \int_{\mathcal{H}_N^3} dXdYdM\ N^2 \sum_{i\geq 1} y^{i-1} \mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)} \\
&\quad \times \mathrm{Tr}\Big(\alpha_i^{(1)} \sum_{k=0}^{i-1} M X^{i-k-1} (X - y^{-1/2}\Lambda_1) X^k M Y^i \\
&\qquad + \alpha_i^{(2)} \sum_{k=0}^{i-1} M Y^i M X^{i-k-1} (X - y^{-1/2}\Lambda_1) X^k \\
&\qquad + \beta_i^{(1)} \sum_{k=0}^{[i/2]-1} \big(M(XY)^{[i/2]-k-1} (X - y^{-1/2}\Lambda_1) Y (XY)^k X^{i\%2} M Y^{i\%2} (XY)^{[i/2]} \\
&\qquad\quad + M(XY)^{[i/2]} (X - y^{-1/2}\Lambda_1) M Y (XY)^{[i/2]} (i\%2) \\
&\qquad\quad + M(XY)^{[i/2]} X^{i\%2} M Y^{i\%2} (XY)^{[i/2]-k-1} (X - y^{-1/2}\Lambda_1) Y (XY)^k \big) \\
&\qquad + \beta_i^{(2)} \sum_{k=0}^{[i/2]-1} \big(M(YX)^{[i/2]-k-1} Y (X - y^{-1/2}\Lambda_1) (YX)^k Y^{i\%2} M X^{i\%2} (YX)^{[i/2]} \\
&\qquad\quad + M(YX)^{[i/2]} Y M (X - y^{-1/2}\Lambda_1) (YX)^{[i/2]} (i\%2) \\
&\qquad\quad + M(YX)^{[i/2]} Y^{i\%2} M X^{i\%2} (YX)^{[i/2]-k-1} Y (X - y^{-1/2}\Lambda_1) (YX)^k \big) \Big)
\end{aligned}
$$

Exchanging the role of (X,Λ_1) and (Y,Λ_2), we find

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}_i^{(1)},\boldsymbol{\beta}_i^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3}dXdYdM\ N^2\sum_{i\geq 1}y^{i-1}\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)}\\
&\quad\times\mathrm{Tr}\Big(\alpha_i^{(1)}\sum_{k=0}^{i-1}MX^iMY^{i-k-1}(Y-y^{-1/2}\Lambda_2)Y^k\\
&\qquad+\alpha_i^{(2)}\sum_{k=0}^{i-1}MY^{i-k-1}(Y-y^{-1/2}\Lambda_2)Y^kMX^i\\
&\qquad+\beta_i^{(1)}\sum_{k=0}^{[i/2]-1}\big(M(XY)^{[i/2]-k-1}X(Y-y^{-1/2}\Lambda_2)(XY)^kX^{i\%2}MY^{i\%2}(XY)^{[i/2]}\\
&\qquad\quad+M(XY)^{[i/2]}XM(Y-y^{-1/2}\Lambda_2)(XY)^{[i/2]}(i\%2)\\
&\qquad\quad+M(XY)^{[i/2]}X^{i\%2}MY^{i\%2}(XY)^{[i/2]-k-1}X(Y-y^{-1/2}\Lambda_2)(XY)^k\big)\\
&\qquad+\beta_i^{(2)}\sum_{k=0}^{[i/2]-1}\big(M(YX)^{[i/2]-k-1}(Y-y^{-1/2}\Lambda_2)X(YX)^kY^{i\%2}MX^{i\%2}(YX)^{[i/2]}\\
&\qquad\quad+M(YX)^{[i/2]}(Y-y^{-1/2}\Lambda_2)MX(YX)^{[i/2]}(i\%2)\\
&\qquad\quad+M(YX)^{[i/2]}Y^{i\%2}MX^{i\%2}(YX)^{[i/2]-k-1}(Y-y^{-1/2}\Lambda_2)X(YX)^k\big)\Big)
\end{aligned}
$$

Finally, we compute the derivative with respect to y to find

$$
\begin{aligned}
&\frac{\partial}{\partial y}Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}_i^{(1)},\boldsymbol{\beta}_i^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3}dAdBdM\ \frac{N}{2}\sum_{i\geq 1}y^{i-1}\\
&\quad\times\mathrm{Tr}\Bigg[\alpha_i^{(1)}\sum_{k=0}^{i-1}\Big(M(A+y^{-1/2}\Lambda_1)^kA(A+y^{-1/2}\Lambda_1)^{i-k-1}M(B+y^{-1/2}\Lambda_2)^i\\
&\qquad\qquad+M(A+y^{-1/2}\Lambda_1)^iM(B+y^{-1/2}\Lambda_2)^kB(B+y^{-1/2}\Lambda_2)^{i-k-1}\Big)\\
&\qquad+\alpha_i^{(2)}\sum_{k=0}^{i-1}\Big(M(B+y^{-1/2}\Lambda_2)^kB(B+y^{-1/2}\Lambda_2)^{i-k-1}M(A+y^{-1/2}\Lambda_1)^i\\
&\qquad\qquad+M(B+y^{-1/2}\Lambda_2)^iM(A+y^{-1/2}\Lambda_1)^kA(A+y^{-1/2}\Lambda_1)^{i-k-1}\Big)\\
&\qquad+\beta_i^{(1)}\sum_{k=0}^{[i/2]-1}\Big(M\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^k\big(A(B+y^{-1/2}\Lambda_2)+(A+y^{-1/2}\Lambda_1)B\big)\\
&\qquad\qquad\times\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]-k-1}\\
&\qquad\qquad\times(A+y^{-1/2}\Lambda_1)^{i\%2}M(B+y^{-1/2}\Lambda_2)^{i\%2}\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&\qquad\quad+M\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\big(AM(B+y^{-1/2}\Lambda_2)+(A+y^{-1/2}\Lambda_1)MB\big)
\end{aligned}
$$

$$
\begin{aligned}
&\times \big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}(i\%2)\\
&+ M\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}(A+y^{-1/2}\Lambda_1)^{i\%2}M(B+y^{-1/2}\Lambda_2)^{i\%2}\\
&\times \big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{k}\big(A(B+y^{-1/2}\Lambda_2)+(A+y^{-1/2}\Lambda_1)B\big)\\
&\times \big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]-k-1}\Big)\\
&+\beta_i^{(2)}\sum_{k=0}^{[i/2]-1}\Big(M\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{k}\big(B(A+y^{-1/2}\Lambda_1)+(B+y^{-1/2}\Lambda_2)A\big)\\
&\times \big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]-k-1}\\
&\times (B+y^{-1/2}\Lambda_2)^{i\%2}M(A+y^{-1/2}\Lambda_1)^{i\%2}\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\\
&+ M\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\big(BM(A+y^{-1/2}\Lambda_1)+(B+y^{-1/2}\Lambda_2)MA\big)\\
&\times \big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}(i\%2)\\
&+ M\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}(B+y^{-1/2}\Lambda_2)^{i\%2}M(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\times \big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{k}\big(B(A+y^{-1/2}\Lambda_1)+(B+y^{-1/2}\Lambda_2)A\big)\\
&\times \big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]-k-1}\Big)\Big]\\
&\times \mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)}\\
&=\frac{1}{2N}\left(\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)Z_N(y;\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i,\boldsymbol{r_{ij}}).
\end{aligned}
$$

□

- Page 65, definition 6. The partition function should read;

$$
\begin{aligned}
&Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{i,j;p}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)}\int_{\mathcal{H}_N^3}dAdBdM\,\exp\Big[-N\mathrm{Tr}\Big(AB+\frac{1}{2}M^2\\
&-\sum_{i\geq 0}y^i\eta\Big\{\\
&+\alpha_i^{(1)}(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^i(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^i\\
&+\alpha_i^{(2)}(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^i(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^i\\
&+\beta_i^{(1)}(M+\eta^{-1/2}\Lambda)\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&\quad\times(A+y^{-1/2}\Lambda_1)^{i\%2}(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^{i\%2}\\
&\quad\times\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^{[i/2]}\\
&+\beta_i^{(2)}(M+\eta^{-1/2}\Lambda)\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\\
&\quad\times(B+y^{-1/2}\Lambda_2)^{i\%2}(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^{i\%2}\\
&\quad\times\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^{[i/2]}\Big\}\Big)\Big]
\end{aligned}
$$

$$= \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)} \int_{\mathcal{H}_N^3} dAdBdM \ \mathrm{e}^{-N\mathrm{Tr}V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)}.$$

- The proof for theorem 4.9 should be changed accordingly.

Proof. The first equation is proven in the same way as the previous model (i.e. $\Lambda = 0$). Here we focus on the proof of the second equation.

Consider the derivative with respect to Λ;

$$\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda^2} Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij};p}) \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)} \int_{\mathcal{H}_N^3} dXdYdT\ N^2 \sum_{i\geq 0} y^i \mathrm{e}^{-N\mathrm{Tr}W_{y,\eta,\alpha,\beta}(X,Y,T;\Lambda_1,\Lambda_2,\Lambda)} \\
&\quad \times \mathrm{Tr}\Big\{ \alpha_{\mathrm{i}}^{(1)}\big((\mathrm{T}-\eta^{-1/2}\Lambda)\mathrm{X}^{\mathrm{i}}\mathrm{T}\mathrm{Y}^{\mathrm{i}} + \mathrm{T}\mathrm{X}^{\mathrm{i}}(\mathrm{T}-\eta^{-1/2}\Lambda)\mathrm{Y}^{\mathrm{i}}\big) \\
&\qquad + \alpha_i^{(2)}\big((T-\eta^{-1/2}\Lambda)Y^iTX^i + TY^i(T-\eta^{-1/2}\Lambda)X^i\big) \\
&\qquad + \beta_i^{(1)}\big((T-\eta^{-1/2}\Lambda)(XY)^{[i/2]}X^{i\%2}TY^{i\%2}(XY)^{[i/2]} \\
&\qquad\quad + T(XY)^{[i/2]}X^{i\%2}(T-\eta^{-1/2}\Lambda)Y^{i\%2}(XY)^{[i/2]}\big) \\
&\qquad + \beta_i^{(2)}\big((T-\eta^{-1/2}\Lambda)(YX)^{[i/2]}Y^{i\%2}TX^{i\%2}(YX)^{[i/2]} \\
&\qquad\quad + T(YX)^{[i/2]}Y^{i\%2}(T-\eta^{-1/2}\Lambda)X^{i\%2}(YX)^{[i/2]}\big)\Big\},
\end{aligned}$$

where $T = M + \eta^{-1/2}\Lambda$ and

$$\begin{aligned}
&W_{y,\eta,\alpha,\beta}(X,Y,T;\Lambda_1,\Lambda_2,\Lambda) \\
&= (X-y^{-1/2}\Lambda_1)(Y-y^{-1/2}\Lambda_2) + \frac{1}{2}(T-\eta^{-1/2}\Lambda)^2 \\
&\quad - \sum_{i\geq 0} y^i(\alpha_i^{(1)}TX^iTY^i + \alpha_i^{(2)}TY^iTX^i) \\
&\quad - \sum_{i\geq 0} y^i\big(\beta_i^{(1)}T(XY)^{[i/2]}X^{i\%2}TY^{i\%2}(XY)^{[i/2]} \\
&\qquad + \beta_i^{(2)}T(YX)^{[i/2]}Y^{i\%2}TX^{i\%2}(YX)^{[i/2]}\big).
\end{aligned}$$

The derivative with respect to η is given by

$$\begin{aligned}
&\frac{\partial}{\partial\eta} Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij};p}) \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N^3)} \int_{\mathcal{H}_N^3} dAdBdM\ \frac{N}{2} \sum_{i\geq 0} y^i \mathrm{e}^{-N\mathrm{Tr}V_{y,\zeta,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2,\Lambda)} \\
&\quad \times \mathrm{Tr}\Big\{ \alpha_i^{(1)}\big(M(A+y^{-1/2}\Lambda_1)^i(M+\eta^{-1/2}\Lambda)(B+y^{-1/2}\Lambda_2)^i \\
&\qquad + (M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^iM(B+y^{-1/2}\Lambda_2)^i\big) \\
&\qquad + \alpha_i^{(2)}\big(M(B+y^{-1/2}\Lambda_2)^i(M+\eta^{-1/2}\Lambda)(A+y^{-1/2}\Lambda_1)^i
\end{aligned}$$

$$
\begin{aligned}
&\quad + (M + \eta^{-1/2}\Lambda)(B + y^{-1/2}\Lambda_2)^i M (A + y^{-1/2}\Lambda_1)^i \big) \\
&+ \beta_i^{(1)} \Big(M \big((A + y^{-1/2}\Lambda_1)(B + y^{-1/2}\Lambda_2) \big)^{[i/2]} \\
&\quad \times (A + y^{-1/2}\Lambda_1)^{i\%2} (M + \eta^{-1/2}\Lambda)(B + y^{-1/2}\Lambda_2)^{i\%2} \\
&\quad \times \big((A + y^{-1/2}\Lambda_1)(B + y^{-1/2}\Lambda_2) \big)^{[i/2]} \\
&\quad + (M + \eta^{-1/2}\Lambda) \big((A + y^{-1/2}\Lambda_1)(B + y^{-1/2}\Lambda_2) \big)^{[i/2]} \\
&\quad \times (A + y^{-1/2}\Lambda_1)^{i\%2} M (B + y^{-1/2}\Lambda_2)^{i\%2} \\
&\quad \times \big((A + y^{-1/2}\Lambda_1)(B + y^{-1/2}\Lambda_2) \big)^{[i/2]} \Big) \\
&+ \beta_i^{(2)} \Big(M \big((B + y^{-1/2}\Lambda_2)(A + y^{-1/2}\Lambda_1) \big)^{[i/2]} \\
&\quad \times (B + y^{-1/2}\Lambda_2)^{i\%2} (M + \eta^{-1/2}\Lambda)(A + y^{-1/2}\Lambda_1)^{i\%2} \\
&\quad \times \big((B + y^{-1/2}\Lambda_2)(A + y^{-1/2}\Lambda_1) \big)^{[i/2]} \\
&\quad + (M + \eta^{-1/2}\Lambda) \big((B + y^{-1/2}\Lambda_2)(A + y^{-1/2}\Lambda_1) \big)^{[i/2]} \\
&\quad \times (B + y^{-1/2}\Lambda_2)^{i\%2} M (A + y^{-1/2}\Lambda_1)^{i\%2} \\
&\quad \times \big((B + y^{-1/2}\Lambda_2)(A + y^{-1/2}\Lambda_1) \big)^{[i/2]} \Big) \Big\}
\end{aligned}
$$

Comparing these two results, we obtain the heat equation (51). □

- Page 92, 2nd line from the top. "Different values of a in the combined score function (57) only had...".
- Page 92, 8th line from the top. "algorithm using 4, 5, and 6 % for pre-selection."
- Page 96, 16th line from the bottom. "..by the model as the best, and the true best candidate."
- Page 104, 10th line from the top. "One of the classical descriptors of protein local structures...".
- Page 118, 8th line from the bottom. "When we consider more practical applications,".

Acknowledgements

First and foremost, I would like to thank my supervisor, Jørgen Ellegaard Andersen, for his help and encouragement throughout the project. I would not have been able to complete my PhD studies without his support through some difficult periods. His drive and optimism have been truly inspirational. I would also like to thank Gergely Bérczi for taking over the official supervising role at a short notice, and his support in the final phase of the project.

I thank my colleagues at the Department of Mathematics, Aarhus University, in particular the members of now-closed QGM. A special thanks to my PhD and office colleagues, Andreas Skovbakke, Lukas Engberg and Simone Siclari for their friendship and generally putting up with me. I also thank Jane Jamshidi for all her help with all administrative and practical matters. I am grateful to Rasmus Villemoes for his help in getting me started and in particular with programming matters. I am also grateful to Jakob Toudahl Nielsen for his help in the preparation of protein data. I would also like to thank the SDU eScience Centre and the Centre for Scientific Computing in Aarhus for providing computing resources for the project.

I am extremely grateful to Hiroyuki Fuji for his generosity in hosting me on two occasions, and for many stimulating discussions. His dedication and kindness have been inspiring. I am grateful to Tsukasa Tada for hosting me at iTHEMS, Riken. I am also grateful to iTHEMS at Riken, and Kagawa University for their generosity in hosting me.

I wish to thank my children, Maika and Aki, for their belief in me and for being there. Last but not least, I wish to thank my wife Bente, for her support and understanding throughout the project and my years at university. I am truly fortunate to have her as my "better half".

Abstract

This thesis presents the results from my PhD studies at Aarhus University, supervised by Professor Jørgen Ellegaard Andersen. The main focus of the PhD project was to investigate applications of the fatgraph model of proteins, which was first proposed by Andersen and others [72]. The studies are exploratory in nature, but are designed with the intention of contributing to the protein folding problem using the fatgraph model.

In the first part of the thesis, a review of mathematical objects and theories related to the project is presented. It is followed by a review of the works utilising fatgraphs in the study of another biological macromolecule, RNA. We review the recursion relations obtained by the so-called cut and join method, matrix models and topological recursion.

In the second part of the thesis, we present new results in relation to the study of protein structures. First the basic fatgraph model of proteins is introduced, and recursion relations for the protein diagrams, obtained by cut and join method, are presented. We construct matrix models that encode generating functions for protein diagrams, and derive partial differential equations, which express the cut and join equations. We then discuss three experimental studies in applications of fatgraph models. In the first project, we introduce a novel model of proteins, which we call protein metastructures, and an associated topological model, which is a modification of the basic fatgraph model. These are used to study β-sheet topology of proteins, which is the configuration of β-strands in β-sheets. We show that the proteins favour less complex β-sheet topologies by comparing the data from the actual proteins and simulated data. Some applications of the models are presented, including an example for combining the method with an existing program for predicting β-sheet topology. As a result, prediction accuracy was improved by 8 percentage points in Precision and 3 percentage points in Recall. The second project takes inspiration from CASP assessment of model quality, and attempts to select the best structure from a set of candidate structures, which aim to reproduce the target protein structure from its primary sequence. We show the topological information contained in our model is enough to predict, if not the best, a structure close to the best candidate structure. The third project aims to predict local geometry of the proteins, expressed as a rotation between peptide units (expressed as an element in the rotation group SO(3)) that are connected by a hydrogen bond, from their topology. The topological information is expressed as a pattern of other hydrogen bonds around the bond in question. We show that the rotation can be predicted to a high accuracy; close to 90% of the predictions lie within a ball centred at the true rotation occupying 1% of the SO(3) space. We conclude the thesis by a brief discussion of potential future challenges and benefits of the use of fatgraph models in the protein structure research.

Resumé

Vi præsenterer resultaterne fra mit PhD studie på Aarhus Universitet, under vejledning af Professor Jørgen Ellegaard Andersen. Målet for PhD projektet var, at undersøge anvendelsesmulighederne for fatgraf-modeller af proteiner, som først foreslået af Andersen, sammen med Penner, Knudsen og Wiuf [72]. Projektet er eksplorativt, men det er vores hensigt at bidrage til foldningsproblemet for proteiner ved at bruge fatgraf-modellerne.

I første del af afhandlingen præsenteres der en redegørelse af de matematiske objekter og teorier, som indgår i projektet. Derefter følger en redegørelse af studier, der har anvendt fatgrafer i en anden biologisk makromolekyle; RNA. Vi beskriver en rekursion for RNA stukturer, opnået ved såkaldt cut-and-join metode, matrixmodel og topologisk rekursion.

I anden del præsenteres der nye resultater i forbindelse med proteinstruktuerer. Vi introducerer den grundlæggende fatgraf-model for proteiner, og derefter anvender vi cut-and-join metoden til at finde rekursionsrelationer for proteindiagrammer. Vi konstruerer matrixmodeller, som koder genererende funktioner for proteindiagrammer, og udleder partielle differentialligninger, som udtrykker cut-and-join ligningerne. Derefter diskuterer vi tre eksperimentelle studier af anvendelse af fatgrafmodeller. I det første projekt introducerer vi en ny og original model for proteiner, som vi kalder proteinmetastrukturer, samt en dertilhørende topologisk model, som er en modificering af den grundlæggende fatgraf model. Vi bruger disse for at studere proteiners β-ark topologi, som er konfigurationen af β-strenge i β-foldede ark. Vi viser, at proteinerne foretrækker mindre komplekse β-ark topologier, ved at sammenligne data fra reelle proteiner og simulerede data. Anvendelser af modellerne præsenteres, samt et eksempel på at kombinere metoden med et eksisterende software-program for forudsigelsen af β-ark topologi. Vi forbedrede forudsigelsesnøjagtigheden med 8 procentpoint i præcisionen og 3 procentpoint i sensitiviteten. Det andet projekt er inspireret af CASP "assessment of model quality". Målet er at udvælge den bedste struktur fra en gruppe kandidatstrukturer, som alle forsøger at gengive strukturen af "target" proteinet ud fra dets primærsekvens. Vi viser, at den topologiske information, som er indeholdt i vores model, er nok til at forudsige, om ikke den bedste, så en struktur tæt på den bedste kandidatstruktur. Det tredie projekts mål er at forudsige, ud fra et proteins topologi, dets lokale geometri, repræsenteret i en rotation mellem to peptideenheder forbundet med en hydrogenbinding. Den topologiske information er angivet som et møster af andre hydrogenbindinger omkring den binding, der undersøges. Vi viser at rotationen forudsiges med en høj nøjagtighed; næsten 90% af forudsigelserne ligger indenfor en kugle omkring den sande rotation, med volumen svarende til 1% af SO(3)-rum. Afhandlingen afsluttes med en kort diskussion af mulige fremtidige udfordringer og fordele ved anvendelsen af fatgraf-modeller indenfor forskning i proteinstrukturer.

Author Statement

The chapters 2 and 3 are reviews of existing works. The protein model in chapter 4 is by Andersen, Knudsen, Penner and Wiuf [72], while the results in chapter 4.3 are author's own work. The results in chapter 4.4 are in collaboration with Andersen and H. Fuji. The protein metastructure in chapter 5 is based on the idea by Andersen and B. Safnuk (unpublished), and developed with inputs from H. Fuji. The results in chapters 6 and 7 are based on the ideas by Andersen and R. Villemoes. The initial coding for chapters 6 and 7 was by R. Villemoes and J.L. Jensen. All other programming and analyses in the thesis, unless otherwise specified, were done by the author. In accordance with GSNS rules, parts of this thesis were also used in the progress report for the qualifying examination.

Contents

1 Introduction

1.1 Protein and the folding problem

Proteins are large biological molecules, or macromolecules, which are essential to a vast array of biological structures and processes, including, but not limited to, replication of DNA, catalysis of biochemical reactions in cells, transport of molecules within and between cells, and as structural elements in cells and organisms. Chemically, a protein is a chain of amino acid residues, or polypeptide. The polypeptide chain is also called the backbone of a protein. There are 20 so-called standard genetic coded amino acids that form proteins. This sequence of amino acids, which can also be thought of as a finite word in an alphabet with 20 letters, is called the primary structure of a protein. In nature, proteins exist as folded, three-dimensional structure unique to each protein. These three-dimensional structures are called native conformations or tertiary structures. In the folded structures of proteins, there are certain commonly occurring local structures, with α-helices and β-sheets being the two most frequently observed examples [5]. The collection of such local structures in a protein's native conformation is called its secondary structure. In some cases, several protein molecules form a structure, that functions as a single protein complex. Such structures are called the quaternary structure of a protein complex.

It is widely accepted that the function of a protein is largely determined by its three-dimensional folded structure [32], and Anfinsen famously postulated in his Nobel Lecture [16], that in theory, one should be able to determine a protein's structure from its primary sequence alone. This is the so-called protein folding problem, and it has since been an area of major research interest [32]. The potential implications of a solution to the problem are wide-ranging. Not only will it represent a significant advancement in our understanding of molecular biology, but it will also have a far-reaching impact on our understanding of certain diseases thought to be caused by malformed proteins, and drug discovery, such as the recent development of COVID-19 vaccines [49].

There has been much progress over the years towards the solution of the protein folding problem [33, 32], partly driven by the biannual CASP : Critical Assessment of protein Structure Prediction [62, 58] events, and aided by the ever-increasing amount of protein structural data available at the Protein Data Bank (PDB) [20]. With the increase in the available computing power, supercomputers have been built to tackle the problem [4], and in the latest CASP competitions (CASP13; [58] and CASP14; [49]), AlphaFold and AlphaFold 2, deep learning programs developed by a team at DeepMind, have made significant advances [83, 82]. In particular, in the 14th and the latest round of CASP, AlphaFold 2 has achieved prediction accuracy, which is comparable to the accuracy of the experimental methods used to obtain the "correct" structure of proteins [49]. In some ways, therefore, it can be said that we are very close to a solution to the problem. Nevertheless, we are still a long way from being able to consider the problem closed. In particular, the methods based on machine learning/deep learning do not make it easy to model the process of protein folding. It is akin to having a still picture of a folded protein, but not a movie, that shows the entire folding process. While the final image is a highly significant achievement, the true understanding of the protein folding problem is achieved only when we understand the entire folding mechanism of proteins.

1.2 Protein folding problem and fatgraph model of proteins

Andersen, together with others, [72] proposed a model of proteins based on the mathematical object called fatgraphs, which can be thought of as a graph, whose vertices and edges are "fattened" to discs and ribbons. Fatgraphs are objects that originate in mathematics and physics, but in recent years they have been successfully used in the study of RNA structures [48, 93, 92]. Proteins' molecular structures are more complex than that of RNA (for example there are 20 different amino acids as "building blocks" for proteins, as opposed to 4 for RNA), but they also share some similarities, such as their structure in nature as a folded strand. These earlier results, therefore, provide an inspiration for using fatgraphs in the study of protein structures. One advantage of using fatgraphs is that each fatgraph corresponds to a surface, whose genus can be used as a measure of complexity in the structure being modelled. Later in this thesis, following the work by Penner, Knudsen, Wiuf and Andersen [72], we will recall that a protein can naturally be considered as a surface, so in that way fatgraphs are very natural objects to consider when modelling proteins.

In this thesis, we will explore the application of protein models based on fatgraphs to the protein folding problem. Our intended, novel approach to the protein folding problem is in two steps, and relies on an intermediate structure which we call the protein graph structure. It is the structure as presented in [72], and consists of the protein's primary structure together with the hydrogen bonds that form between non-adjacent residues along the polypeptide chain. We describe the protein graph structure in more details in section 4.1, but note here that these hydrogen bonds are an important factor in stabilising the folded structure of a protein [5]. In the first step, we predict a protein's graph structure from its primary structure. Here the idea is to enumerate all possible graph structures from a given primary sequence, and using an appropriate energy function that assigns a pseudoenergy or a"score" to each graph structure, to produce a set of most likely structures. In the second step we use the resulting graphs to predict the tertiary structure. This is done by first predicting the structure locally around each hydrogen bond, then determining the entire tertiary structure that is compatible with the predicted local structures. This approach is inspired by the two previous works of Andersen and others ([74, 73]). In the first paper, they have shown certain topological invariants associated with a protein's graph structure are good descriptors of its three-dimensional structure [74]. In the second, they associated a rotation in $\mathbb{R}^3$ to each hydrogen bond in proteins and showed that they are concentrated around localised clusters, which correspond well to common local structures [73]. The inspiration also comes from earlier works by Andersen, Chekhov, Penner, Reidys, Sułkowski and others, on the structure of another macromolecule, RNA [3, 7, 77, 8, 15, 14, 13]. We will review these earlier studies, together with the construction of the RNA fatgraph model in detail in section 3. We note here, that studying topological structures of RNA molecules filtered by genus has allowed identification of the basic building blocks for these structures, and development of an algorithm to build higher-genus structures using these basic structures ([77, 7]). It also allowed computation of the theoretically possible numbers of certain classes of (simplified) RNA structures [8]. Furthermore, the technique can be adopted to lower levels of abstraction by considering more parameters, thus allowing for

enumeration of more realistic structures [3, 15]. In [3], the lengths of boundary components and the numbers of bivalent (unpaired) vertices in the boundary components are considered to construct corresponding matrix models, which is a technique for enumerating fatgraphs using a formal matrix integral (We will discuss matrix models in more detail in section 2.3). In [15], these two parameters are combined to a unified parameter to produce a single model. In both papers, a method called cut and join is used to solve the matrix model and obtain recursion relations for the number of distinct structures. One advantage of the fatgraph model is that one may utilise the tools developed in mathematics, independent of what is being modelled. This is demonstrated in [14], where the recursion relation is obtained as a series of partial differential equations, and in [13], where the solution is obtained using a technique called topological recursion. One interesting aspect of this enumeration is that the class of RNA structure called shapes are in one-to-one correspondence with the cells in a decomposition of Riemann's moduli space of a surface with one boundary component [8]. Of course, the composition and structure of proteins are different from that of RNA, but these studies provide us with the guiding principle, that the graph structures of proteins can be studied effectively by considering their topology. The RNA model will be adapted to proteins, largely following the procedure described in [72], and adjusted further for particular applications.

1.3 Content of this thesis

This thesis is divided into two parts. In the first part we will provide a detailed review of the theoretical foundations for the project, primarily based around, but not limited to, the earlier work on RNA structures. The topics such as fatgraphs, matrix model and topological recursion will be explored in more generality. Fatgraph is an object first studied in mathematics for indexing a certain decomposition of Riemann's moduli space [71, 88], and in theoretical physics as index sets of large-N limit of certain matrix models [21, 1]. It has been applied in geometry [45, 71], physics [54], and in study of RNA structures [8]. In section 2.1 we will give a mathematical definition of fatgraphs, and describe how we can associate a topological surface to a fatgraph. We also review how fatgraphs can be efficiently stored as a pair of permutations, which is especially convenient for large-scale computations. To demonstrate the historical background of the subject, we then review the theory of Penner and Strebel [70, 88], which states that there is a one-to-one correspondence between cells in the mapping class group invariant cell decompositions of a (decorated) Teichmüller space and homotopy classes of fatgraphs (section 2.2). We will then introduce the theory of matrix integrals as a way of enumerating fatgraphs, filtered by their genera, in section 2.3. The study of the link between matrix integrals and fatgraphs goes back to 1970's, when 't Hooft [1] found that fatgraphs (called planar diagrams in the paper) appear naturally in quantum chromodynamics, when the size of the gauge group $U(N)$ tends to infinity. This has lead to other studies, where matrix models are used as a method for enumerating fatgraphs [25, 21], although there are many more applications of matrix models (see [38] for examples). Next in section 2.4, we describe the theory of topological recursion, which is a powerful framework developed by Chekhov, Eynard and Orantin [28, 43], with application in a number of fields (see [36]), including providing a solution for a matrix model, such as the ones that can be constructed for RNA

structures.

In section 3, we review the application of the above theories in the study of RNA structures, based on the earlier works by Andersen, Chekhov, Penner, Reidys, Sułkowski and others [7, 8, 9, 12, 3, 14, 15, 13]. We will start by a brief review of the molecular structure of RNA, and the construction of the RNA fatgraph model in section 3.1. We will also introduce a number of relevant classes of RNA structures. In section 3.2, based on the materials presented in [3, 15], we demonstrate the so-called cut and join method to find recursion relations for the number of distinct RNA structures, filtered by their topological characteristics. We then construct a matrix model for RNA structures in section 3.3, following [14]. In the final part of section 3, we will describe how the topological recursion framework, introduced in section 2.4 can be applied to the RNA matrix model to obtain the solution.

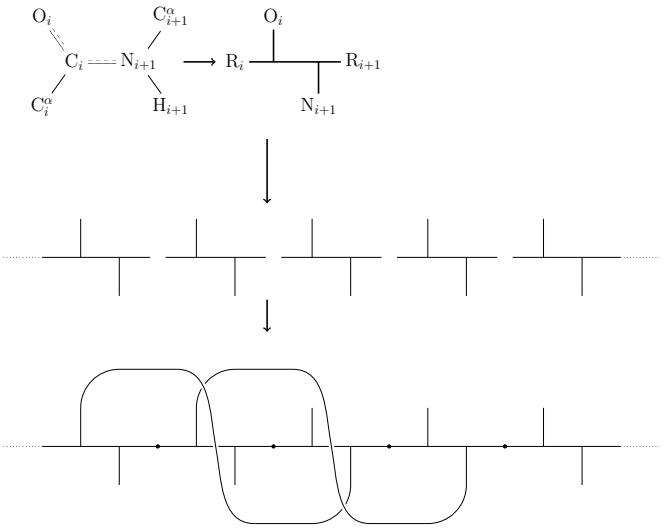


Figure 1: Basic fatgraph model of proteins. Idealised peptide units are glued together, before edges representing the hydrogen bonds are added.

In the second part, we will discuss the main part of this project; the application of the protein fatgraph model to the study of topological and geometric structure of proteins. We begin by giving a brief review of the molecular structure of proteins (section 4.1). We then introduce our basic protein model, as proposed in [72] in section 4.2. It is a model based on the so-called peptide units of proteins, to which hydrogen bonds are added as edges connecting carboxyl oxygens and amino hydrogens (figure 1). Here we will also describe how we can assign an element of the rotation group SO(3) to each of the hydrogen bonds in proteins [72]. A natural question to ask about such fatgraph models is whether there is a recursion relation, similar to that identified for RNA structures in [3, 15] (section 3.2). In section 4.3, we prove a recursion relation for the number $\mathcal{N}_{g,k,l}(\boldsymbol{b};\boldsymbol{q})$ oriented, multi-backbone protein diagrams by the so-called cut and join method. Here g, k, l denote the genus, the number of hydrogen bond edges, and the number of unbonded edges, respectively. $\boldsymbol{b} = (b_0, b_1, \dots)$ records the number b_i of backbones with precisely i paired or unpaired vertices, and

$\boldsymbol{q} = (q_{\boldsymbol{p}_K}, \dots)$ is the sequence of the numbers $q_{\boldsymbol{p}_K}$, indexed by another sequence $\boldsymbol{p}_K$, which records the sequence of unpaired carboxyl oxygens denoted by the letter "O", and amino hydrogens denoted by the letter "N" (see section 4.3 for detailed descriptions).

Theorem. *$\mathcal{N}_{g,k,l}(\boldsymbol{b};\boldsymbol{q})$ satisfies the following recursion relation;*

$$
\begin{aligned}
k\mathcal{N}_{g,k,l}(\boldsymbol{b};\boldsymbol{q}) &= \sum_{K\geq 1}\sum_{\boldsymbol{p}_K}(q_{\boldsymbol{p}_K}+1)\sum_{\substack{1\leq I\leq K\\ p_I\neq p_J}}\sum_{I<J\leq K}\mathcal{N}_{g,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+s_{I,J}(\boldsymbol{p}_K))\\
&+\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L})\\
&\qquad\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L}\mathcal{N}_{g-1,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L))\\
&+\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}\sum_{g_1+g_2=g}\sum_{k_1+k_2=k-1}\sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}}\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L}\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}}\\
&\qquad q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{N}_{g_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{N}_{g_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)}),
\end{aligned}
$$

where $b^{(a)} = \sum_{i=1}^{\infty} b_i^{(a)}$, $a = 1, 2$ and

$$
\begin{aligned}
s_{I,J}(\boldsymbol{p}_K) &= \boldsymbol{e}_{\boldsymbol{p}_K} - \boldsymbol{e}_{(p_1\dots p_{I-1}p_{J+1}\dots p_K)} - \boldsymbol{e}_{(p_{I+1}p_{I+2}\dots p_{J-2}p_{J-1})}\\
t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L) &= \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} - \boldsymbol{e}_{(p_1\dots p_{I-1}r_{J+1}\dots r_L r_1\dots r_{J-1}p_{I+1}\dots p_K)}.
\end{aligned}
$$

We will also prove the corresponding result for the number $\mathcal{M}_{h,k,l}(\boldsymbol{b};\boldsymbol{q})$ of orientable and non-orientable protein diagrams, where h is twice the genus if the diagram is orientable, and the number of cross-caps if not.

Theorem. *$\mathcal{M}_{h,k,l}(\boldsymbol{b};\boldsymbol{q})$ satisfies the following relation.*

$$
\begin{aligned}
k\mathcal{M}_{h,k,l}(\boldsymbol{b};\boldsymbol{q}) = & \\
& \sum_{K\geq 1}\sum_{\boldsymbol{p}_K}(q_{\boldsymbol{p}_K}+1)\sum_{\substack{1\leq I\leq K\\ p_I\neq p_J}}\sum_{I<J\leq K}\big[2\mathcal{M}_{h,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+s_{I,J}(\boldsymbol{p}_K)) \\
& \quad +\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+u_{I,J}(\boldsymbol{p}_K))\big] \\
& +\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L}) \\
& \times\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L}\big[\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{e}_{2\boldsymbol{k}+\boldsymbol{l}};\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)) \\
& \quad +\mathcal{M}_{h-2,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L))\big] \\
& +\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}\sum_{h_1+h_2=h}\sum_{k_1+k_2=k-1}\sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}}\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L} \\
& \Bigg[\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}} q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{M}_{h_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{M}_{h_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)}) \\
& +\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}} q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{M}_{h_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{M}_{h_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)})\Bigg],
\end{aligned}
$$

where $s_{I,J}$ and $t_{I,J}$ are as above, and

$$
\begin{aligned}
u_{I,J}(\boldsymbol{p}_K) &= \boldsymbol{e}_{\boldsymbol{p}_K}-\boldsymbol{e}_{(p_1\cdots p_{I-1}p_{J-1}p_{J-2}\cdots p_{I+1}p_{J+1}\cdots p_K)}, \\
\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L) &= \boldsymbol{e}_{\boldsymbol{p}_K}+\boldsymbol{e}_{\boldsymbol{r}_L}-\boldsymbol{e}_{(p_1\ldots p_{I-1}r_{J-1}\ldots r_1r_L\ldots r_{J+1}p_{I+1}\cdots p_K)}.
\end{aligned}
$$

We also invesigate the enumeration problem using matrix model. By using a combinatorial parameter $\boldsymbol{\ell_{ij}}$ that records the unpaired vertices similarly to $\boldsymbol{q}$ introduced above, we construct the protein 2-matrix model which gives the generating function for the number of protein diagrams of a certain type, where restrict the connection along the backbone to either untwisted or twisted, but not mixed (figure 2 and figure 3).

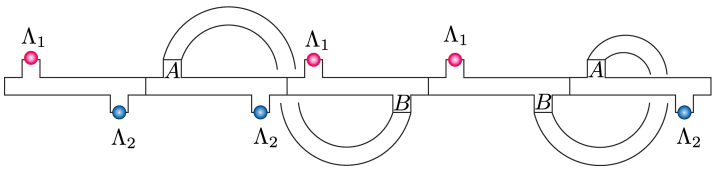


Figure 2: A protein diagram with an untwisted backbone.

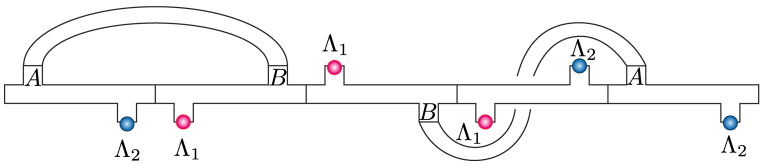


Figure 3: A protein diagram with a twisted backbone.

Theorem. *Let* $Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ *denote the exponential of the generating function:*

$$Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) = \exp\left[F(1/N,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\right].$$

$Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ *is given as the partition function of the hermitian 2-matrix model with external fields* Λ_1 *and* Λ_2*:*

$$\begin{aligned}
&Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) \\
&= \frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}} dAdB\ \exp\Bigg[-N\mathrm{Tr}\Bigg(AB - \sum_{i\geq 0}\alpha_i y^i (A+y^{-1/2}\Lambda_1)^i (B+y^{-1/2}\Lambda_2)^i \\
&\qquad\qquad - \sum_{i\geq 0}\beta_i y^i ((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^i\Bigg)\Bigg] \\
&= \frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}} dAdB\ \mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)},
\end{aligned}$$

where $\mathrm{Vol}_N = N^{N(N+1)/2}\mathrm{Vol}(\mathcal{H}_N)$*, and* $r_{\boldsymbol{ij}}$*'s are defined by the single trace for* Λ_1 *and* Λ_2*'s such as*

$$r_{(i_1,\ldots,i_K),(j_1,\ldots,j_K)} = \frac{1}{N}\mathrm{Tr}(\Lambda_1^{i_1}\Lambda_2^{j_1}\Lambda_1^{i_2}\Lambda_2^{j_2}\cdots\Lambda_1^{i_K}\Lambda_2^{j_K}).$$

We then prove the generating function obeys the heat equation, which can also be expressed as cut and join equation.

Theorem. *The generating function* $Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ *satisfies the heat equation:*

$$\begin{aligned}
&\frac{\partial}{\partial y}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) \\
&= \frac{1}{2N}\left(\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2} + \mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}),
\end{aligned}$$

where $\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}$ *denotes*

$$\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2} = \sum_{a,b=1}^{N}\frac{\partial^2}{\partial\Lambda_{1ab}\partial\Lambda_{2ba}}.$$

We extend the model further by introducing another matrix and then another external matrix, to find the model for the diagrams, where the two types of

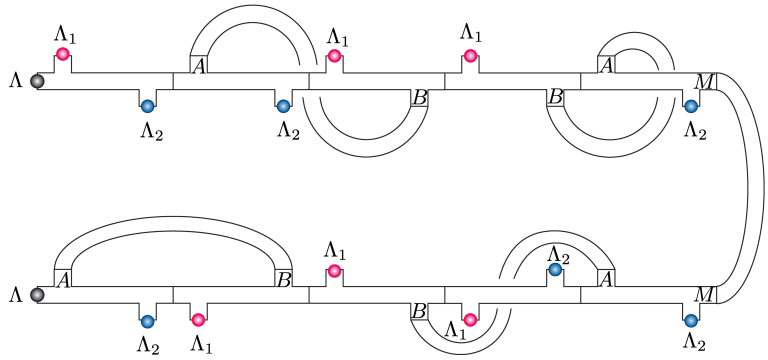


Figure 4: An extended proten diagram.

backbones can be connected to each other (figure 4). It is given by the hermitian 3-matrix model;

$$\begin{aligned} &Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{i,j;p}}) \\ &\quad = \frac{1}{\mathrm{Vol}(\mathcal{H}_N)^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\ \mathrm{e}^{-N\mathrm{Tr}V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)}, \end{aligned}$$

where the potential $V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)$ is given by;

$$\begin{aligned} &V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda) \\ &\quad = AB+\frac{1}{2}M^2-\sum_{i\geq 0} y^i\eta(M+\eta^{-1/2}\Lambda)\Bigg\{ \\ &\qquad +\alpha_i^{(1)}(A+y^{-1/2}\Lambda_1)^i(B+y^{-1/2}\Lambda_2)^i \\ &\qquad +\alpha_i^{(2)}(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^i \\ &\qquad +\beta_i^{(1)}\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^i \\ &\qquad +\beta_i^{(2)}\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^i\Bigg\}(M+\eta^{-1/2}\Lambda)\Bigg). \end{aligned}$$

We will prove $Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{i,j;p}})$ also obeys the heat equation;

Theorem. *The partition function* $Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})$ *obeys heat equations:*

$$\left(\frac{\partial}{\partial y}-\frac{1}{2N}\left(\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)\right)Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})=0, \tag{1}$$

$$\left(\frac{\partial}{\partial\eta}-\frac{1}{2N}\frac{\partial^2}{\partial\Lambda^2}\right)Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})=0. \tag{2}$$

One of the important research areas within the protein folding problem is the identification of β-sheets, both in terms of which parts of the backbone participate in β-sheet structures and how these segments are configured within each β-sheet. We introduce a new structure, which we call protein metastructure, to model β-sheet configurations in proteins (section 5.2). This is a simplification of protein β-sheet structures, yet retains essential information to understand the configuration of β-sheets. Furthermore, there is a natural fatgraph structure associated with each protein metastructure, which enables the study of topological invariants in β-sheet structures (section 5.3). In section 5.4, we will show that a class of protein metastructures, which we call the extended metastructures, satisfy a the same recursion relation relation for RNA diagrams, which was discovered by Harer and Zagier [45] and recalled in [8]. By comparing the topological characteristics of proteins from the existing database and simulated protein structures, we discover that the real protein metastructures tend to have lower genera than simulated structures (section 5.5). This may be an indication that when protein folds, more complex (i.e. higher genus) structures are energetically unfavourable. We then describe a series of experiments carried out as potential applications of protein metastructures, all based around prediction of metastructure (section 5.7). In the first experiment, we use topological invariants of protein metastructures to classify "candidate" structures generated for a target protein (section 5.7.1). The quality of predicted structures is measured in Recall and Precision, given by

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned}$$

where TP, FP and FN stand for the number of true positive, false positive, and false negative strand pairings. We find that our filter based on metastructure topology rejects more than 90% of candidate structures, while retaining the high-quality candidates (table 1). This effect became more pronounced when we removed four (out of 189) proteins, which accounted for more than 99% of candidate structures (table 2).

Quality	Precision		Recall	
Low	7.38%	(4850914/65738106)	7.42%	(4819915/64953828)
Medium	7.88%	(82780/1050828)	6.19%	(112857/1824369)
High	89.42%	(93/104)	9.36%	(1015/10841)

Table 1: Acceptance rate and number of acceptance (in parentheses, {# accepted}/{#candidates}) by quality classes. The topology filter effectively reduces the number of candidate structures, while retaining high-quality structures.

In the second experiment, we try to predict a metastructure for a given protein by using a sequence alignment and topological invariants (section 5.7.2). Finally, we present another prediction method, which utilises an established β-sheet prediction programme called Betapro [29] and topological invariants of metastructures (section 5.7.3). We show that by utilising the metastructure

Quality	Precision		Recall	
Low	15.18%	(13136/86540)	21.27%	(11729/55150)
Medium	9.07%	(9929/109416)	7.86%	(11042/140460)
High	91.18%	(93/102)	86.38%	(387/448)

Table 2: Acceptance rate and number of acceptance (in parentheses, {# accepted}/{#candidates}) by quality classes, excluding four proteins with > 5000 candidate structures. The acceptance rates for high-quality structures are improved further.

topology, we can improve the prediction accuracy of an existing software programme.

Program	Precision	Recall
Betapro [29]	0.59	0.54
Current Study	0.67	0.57

Table 3: We were able to improve the prediction accuracy of Betapro programme.

A major driving force behind the advancement in the protein folding problem has been the biannual series of community-wide experiments called CASP: Critical Assessment of protein Structure Prediction [62]. In each round of CASP, a set of protein primary structures are published, while their secondary and tertiary structures are kept secret. The participants' goal is to predict the tertiary structures of these proteins. CASP includes model accuracy estimate as one of the sub-categories, where one attempts to estimate the quality of the submitted entries. Inspired by this, we designed algorithms for model quality estimation, which uses the primary structure and information about the hydrogen bonds of the target proteins, instead of the primary structure alone. The project is presented in section 6. We begin the section by reviewing the primary metric called GDT (Global Distance Test), which CASP uses to measure the closeness of two protein tertiary structures (section 6.1), before briefly describing the dataset used for this project (section 6.2). We then describe the first of our two algorithms, which is a linear regression method with information about the hydrogen bonds as independent variables (section 6.3). The other algorithm attempts to mimic the GDT algorithm, but based only on the two proteins' hydrogen bond structures (section 6.4). We show that these relatively simple algorithms, together with the information about the hydrogen bonds which is not available in CASP experiments, can predict the best candidate structures to accuracy levels comparable to those observed in CASP (table 4). We then end the section by presenting a brief discussion of the results from the two algorithms (section 6.5).

In section 7, we shift our focus from the topology to the geometry of proteins, and present our project to investigate the relationship between the local geometric structure of proteins, represented by the rotation along a given hydrogen bond, and the topological structure, given as the primary structure and the hydrogen bonds as a graph. In section 7.1, we define the primary object of our investigation, which we call the H-bond local pattern. We then present two

Linear Regression	graph-GDT	CASP11	CASP12
5.5	19.1	4.6-11	5.0-20

Table 4: Average ΔGDT for the two methods, together with results from all models in CASP10 [55] and CASP12 [56]. ΔGDT is the difference between the highest GDT score among the submitted predictions and the score of the prediction selected by the model.

Range of d		By pattern matching	By pattern alignment
$0 \leq d < 0.2664$	(0.1%)	64.75	51.31
$0.2664 \leq d < 0.4567$	(0.5%)	83.47	69.55
$0.4567 \leq d < 0.5766$	(1.0%)	89.05	76.67
$0.5766 \leq d < 0.7862$	(2.5%)	94.02	84.94
$0.7862 \leq d < 0.9968$	(5.0%)	96.24	89.74
$0.9968 \leq d < 1.2689$	(10.0%)	97.64	93.08
$1.2689 \leq d < 1.7663$	(25.0%)	98.99	96.24
$1.7663 \leq d < 2.3099$	(50.0%)	99.77	98.31
$2.3099 \leq d < 2.7437$	(75.0%)	99.93	99.22
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00

Table 5: Accumulative % of hydrogen bonds whose predicted rotation values lie within the specified distance from the true rotation values. Results from H-bond pattern matching and alignment of H-bond patterns. The numbers in the parentheses show the volume of the ball whose radius is the upper limit of the range, as a proportion of the volume of entire SO(3).

methods, designed to predict the rotation along a given hydrogen bond from the H-bond local pattern around the bond. The first method tries to find an exact match for the given H-bond local pattern in the database constructed from the training dataset (section 7.2), while the second method uses an alignment algorithm to find the H-bond local pattern in the training dataset, that best aligns with the given pattern (section 7.3). We achieved high accuracy in predicting the rotations, particularly with the pattern matching algorithm, where close to 90% of the prediction was within 0.5766 from the true rotations, corresponding to 1% of the volume of SO(3). In each section, we present a detailed description of the method and the results. The section ends with a brief discussion where we compare the results from the two methods (section 7.4).

We end this thesis by a brief discussion, where we compare the results from our three projects on protein structures, and the earlier results from RNA, before presenting some concluding remarks on future prospects.

2 Mathematical foundation for the project

2.1 Fatgraph

Let τ be a finite graph consisting of a set of vertices $V = V(\tau)$ and a set of edges $E = E(\tau)$, where the edges do not contain their endpoints. Construct a set of *half-edges* $H = H(\tau)$ by removing a single point from each edge in E. We have, by construction, $\#H = 2\#E$, where $\#X$ denotes the cardinality of a set X. A half-edge is *incident* on a vertex $u \in V$, if u is contained in its closure. The number of half-edges that are incident on a vertex is called the *valency* of the vertex.

A *fatgraph* τ is a finite graph, together with a cyclic ordering of half-edges incident at each vertex. This ordering of half-edges gives rise to "cycles" of (oriented) edges in the graph as follows. Pick an edge e_1 and choose an orientation for e_1. Let v_1 be the vertex at the end of the oriented edge e_1. We set the next edge in the cycle, e_2, to be the edge immediately following e_1 in the cyclic ordering at v_1, with the orientation pointing away from v_1. At the endpoint of the (oriented) edge e_2, the procedure is repeated until we encounter an (oriented) edge which has already been traversed. If a vertex has a single edge incident upon it, we take the single incident edge, with the orientation pointing away from the vertex, as the next edge. The cycles of oriented edges thus obtained are called boundary cycles.

The above construction is perhaps easier to visualise with an alternative presentation of fatgraphs, which can be obtained by "fattening" the underlying graph τ; i.e. by expanding the vertices to discs, and edges to "ribbons" connecting these discs. More precisely, the construction is done as follows:

For each k-valent vertex $u \in V$ with $k \geq 2$, we associate an oriented surface diffeomorphic to a polygon P_u with $2k$ sides containing a single k-valent vertex in its interior. Each half-edge incident on u is also incident on the mid-point on every other side of P_u. These sides are identified with half-edges incident on u such that the induced counter-clockwise cyclic ordering of the sides of P_u agrees with the cyclic ordering of half-edges at u. For a univalent vertex u, the corresponding surface P_u is diffeomorphic to a 2-gon, with one side containing u and the other side identified with the single half-edge incident on u. The surface $F(\tau)$ associated to τ is the quotient of the disjoined union $\bigsqcup_{u \in V} F_u$, where the sides, oriented with the interior of P_u to the left, are identified by an orientation-reversing homeomorphism that preserve mid-points, if the corresponding half-edges are contained in a common edge of τ. The subgraphs in P_u for $u \in V$ then combine to give a fatgraph embedded in $F(\tau)$, which we identify with τ in the natural way, so we consider $\tau \subseteq F(\tau)$. We thus obtain an oriented surface

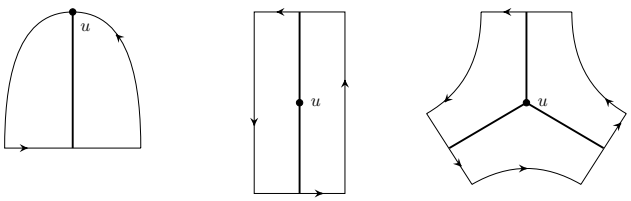


Figure 5: 1-, 2-, and 3-valent vertex u with associated surfaces.

with boundary $F(\tau)$ associated to τ. Note the boundary components of $F(\tau)$ correspond to the boundary cycles defined above (figure 6).

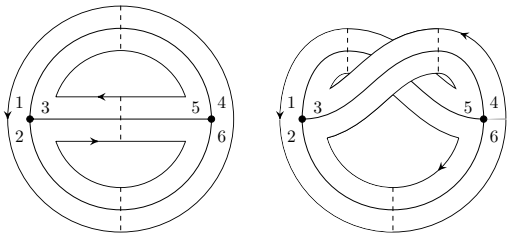


Figure 6: Two fatgraphs with two three-valent vertices, drawn with their associated surfaces.

Let τ be a connected fatgraph, with v vertices, e edges, and r boundary cycles. The *Euler characteristic* of τ is given by

$$\chi(\tau) = v - e + r = 2 - 2g,$$

where g is the genus of τ. Given a connected fatgraph τ with v vertices, e edges and r boundary cycles, we may adjoin one once-punctured disc to each boundary to obtain a surface F_g^r of genus g with r punctures. We then have inclusions $\tau \subset F(\tau) \subset F_g^r$. We call the fatgraph τ a *spine* of $F = F_g^r$ (see figure 10b for an example).

One way of expressing fatgraphs is as a pair of permutations, σ and ι. σ is the product of cycles, given as follows. We first choose one vertex in the graph and number the half-edges around it, then choose the next vertex and number its half-edges, and so on. In this way each k-valent vertex defines a k-cycle and σ is the product of all of them. ι is a product of 2-cycles, each defining which half-edge is connected to which. For example the first fatgraph in figure 6 will have $\sigma = (123)(456), \iota = (14)(26)(35)$, whereas the second graph will have $\sigma = (123)(456), \iota = (15)(26)(34)$. Note cycles in σ and ι are disjoint, so these expressions are independent of the order in which the vertices are chosen.

A useful feature of this expression is that the boundary cycles of a fatgraph τ given by σ, ι are the cycles in $\rho = \sigma \circ \iota$. In this way we can describe the "dual" of a fatgraph given by (σ, ι) by (ρ, ι) (figure 7). An automorphism of a fatgraph given by σ and ι is a bijection ϕ from the set H of half-edges to itself, such that $\sigma = \phi \circ \sigma \circ \phi^{-1}$ and $\iota = \phi \circ \iota \circ \phi^{-1}$. For graphs where all vertices have valency at least three, we have the following elementary lemma, (see e.g. [37]).

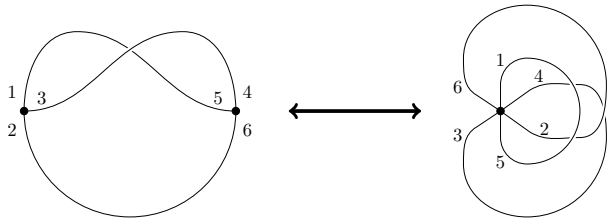


Figure 7: Fatgraph to the left, given by $((123)(456), (15)(26)(34))$ is dual to the fatgraph to the right, given by $((163524), (15)(26)(34))$.

Lemma 2.1. *Let τ be a fatgraph consisting of n_3 three-valent vertices, n_4 four-valent vertices, ..., and n_d d-valent vertices, given by (σ, ι), Let $\#\mathrm{Aut}(\tau)$ be the number of automorphisms, and let $\#\mathrm{Glu}(\tau)$ be the number of "gluings", i.e. the number of ways of gluing together the given $n_3, n_4, \ldots, n_d$ vertices to obtain the graph τ. Then $\#\mathrm{Aut}$ and $\#\mathrm{Glu}$ satisfy*

$$\#\mathrm{Aut} \times \#\mathrm{Glu} = \prod_{j=3}^{d} j^{n_j} n_j!.$$

Proof. Let τ be a fatgraph and define

$$G_\tau = \{\phi \mid \sigma = \phi \circ \sigma \circ \phi^{-1}\}$$

to be the subgroup of the group of bijections from the set of half-edges to itself, which leaves σ invariant. Then

$$\mathrm{Aut}(\tau) = \{\phi \in G_\tau \mid \iota = \phi \circ \iota \circ \phi^{-1}\}$$

is the stabiliser of ι under the G_τ action, and

$$\mathrm{Glu}(\tau) = \{\phi \circ \iota \circ \phi^{-1} \mid \phi \in G_\tau\}$$

is the orbit of ι. By the orbit-stabiliser theorem, we have $\#\mathrm{Aut} \times \#\mathrm{Glu} = \#G_\tau$. There are j permutations for each j-valent vertex which leaves σ invariant, and we may also permute n_j j-valent vertices. The result follows. □

A *metric* on a fatgraph τ is an assignment of some non-negative real number $\mu(e)$ to each edge e of τ, such that there are no cycles in τ whose constituent edges all have vanishing μ value. Note this *no-vanishing cycle condition* ensures that each component of the (possibly empty) forest $\Phi \subset \tau$, consisting of the edges with vanishing μ values, can be contracted to a distinct vertex to produce a fatgraph τ_Φ with a strictly positive metric.

The construction of fatgraphs can be extended to include non-orientable fatgraph, by endowing each edge with an extra binary information, which we refer to as "twisted" and "untwisted". When constructing the associated surface, the identification of the sides corresponding to half-edges is orientation-preserving, if and only if the corresponding edge is twisted. An example of non-orientable fatgraph and the associated surface is given in figure 8. Note the representation of fatgraphs as pairs of permutations, described above, will not work for non-orientable fatgraph. The term *non-oriented fatgraphs* is used to mean a union of orientable and non-orientable fatgraphs.

2.2 The Penner-Strebel decomposition

In this section, we provide a brief summary of the Penner-Strebel theory following the discussion in [70]. The theory proves a deep link between fatgraphs and certain cell decomposition of decorated Teichmüller spaces, and sets the enumeration problems of protein fatgraphs, which we present later in the thesis, in more classical mathematical context.

Consider a three-dimensional real vector space V with a non-degenerate quadratic form $\langle \cdot, \cdot \rangle$, such that the corresponding metric has an expression

$$-\,\mathrm{d}x_0^2 + \mathrm{d}x_1^2 + \mathrm{d}x_2^2.$$

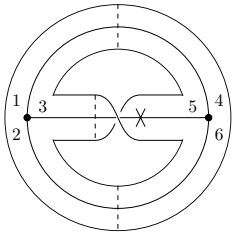


Figure 8: Non-orientable fatgraph and its associated surface. A twisted edge is denoted by an "x".

We call V equipped with this metric *Minkowski three-space* $\mathbb{M}$. The set

$$\{v \in V : \langle v, v\rangle = -1\} = \{x \in \mathbb{M} : -x_0^2 + x_1^2 + x_2^2 = -1\}$$

has two components, and the component $\mathbb{H}$ with $x_0 > 0$ is a model of the hyperbolic plane. The form $\langle \cdot, \cdot \rangle$ restricts to a Riemannian metric on tangent spaces to $\mathbb{H}$. An isometry of $\mathbb{H}$ with the Poincaré disk model is given by the projection onto the unit disk $\mathbb{D} = \{x \in \mathbb{M} : x_0 = 0, x_1^2 + x_2^2 < 1\}$ along the line from $(-1, 0, 0)$ to a point on $\mathbb{H}$. We define the *light-cone* $L \subset \mathbb{M}$ to be

$$L = \{v \in V : \langle v, v\rangle = 0\} = \{x \in \mathbb{M} : x_0^2 = x_1^2 + x_2^2\}$$

and the *positive light-cone* to be

$$L^+ = \{x \in L : x_0 > 0\}.$$

We say a point $x \in \mathbb{M}$ lies "on" L^+ if $x \in L^+$, "inside" L^+ if x lies in the component of $(1, 0, 0)$ of $\mathbb{M} - L$, and "outside" L^+ otherwise. The radial projection from $\mathbb{H}$ to $\mathbb{D}$ extends to the map

$$\bar{\cdot} : \mathbb{H} \cup L^+ \to \mathbb{D} \cup S^1_\infty,$$

where $S^1_\infty = \{x \in \mathbb{M} : x_0 = 0, x_1^2 + x_2^2 = 1\}$ and the map sends the ray on L^+ from the origin to the point where the ray's projection onto the plane $\{x_0 = 0\}$ intersects S^1_∞. We let a point $w = (w_0, w_1, w_2) \in L^+$ correspond to the horocycle

$$h = \{x \in \mathbb{H} : \langle w, x\rangle = -1\}.$$

The centre of $\bar{h}$ is the point $\bar{w}$ and the Euclidean radius of $\bar{h} \subset \mathbb{D}$ is $(1 + w_0)^{-1}$. Hence we see that $\bar{\cdot}$ induces an identification of L^+ with the bundle of horocycles over S^1_∞.

The group of linear isomorphisms of $\mathbb{M}$ preserving the quadratic form is the indefinite orthogonal group $\mathrm{O}(V) = \mathrm{O}(1, 2)$. The subgroup of $\mathrm{O}(1, 2)$ preserving the orientation of V and $\mathbb{H}$ is the component of identity denoted $\mathrm{SO}^+(1, 2)$ and is isomorphic to $\mathrm{PSL}_2(\mathbb{R})$. An element of $\mathrm{PSL}_2(\mathbb{R})$ is called *hyperbolic* (*parabolic*, *elliptic*), if the square of its trace is > 4 ($= 4, < 4$, respectively). The corresponding elements in $\mathrm{SO}^+(1, 2)$ are also called hyperbolic, parabolic and elliptic. Hyperbolic elements of $\mathrm{SO}^+(1, 2)$ have two eigenvectors on L^+, one with the corresponding eigenvalue λ, which is real and positive with $|\lambda| \neq 1$, and the other with λ^{-1}. The third lies outside L^+ with eigenvalue 1. Parabolic

elements have a unique eigenvector on L^+ with eigenvalue 1, and no eigenvector inside L^+. Elliptic elements have all their eigenvalues on the unit circle and one eigenvector inside L^+.

Consider a closed surface F_g of genus g with a set of distinguished points $P = \{x_1, x_2, \ldots, x_s\}$, with $2g - 2 + s > 0$. Let $F_g^s = F_g - P$ be a punctured surface of genus g with s punctures. We will consider the Teichmüller space of F_g^s, which is the space of marked complete hyperbolic structures of finite area on F_g^s.

Definition 1. The Teichmüller space $\mathcal{T}_g^s$ of F_g^s is

$$\mathcal{T}_g^s = \{(M, f)\}/\sim,$$

where M is F_g^s equipped with a complete hyperbolic structure, and $f : F_g^s \to M$ is a homeomorphism, called a *marking*. The relation is given by $(M, f) \sim (N, g)$ if and only if there is an isometry $\imath : M \to N$ such that $\imath \circ f$ is homotopic to g.

Note because of the completeness of the metric, each puncture gives rise to a cusp. It will be useful to give an equivalent definition of $\mathcal{T}_g^s$ as a set of equivalence classes of representations.

Definition 2. The Teichmüller space $\mathcal{T}_g^s$ of F_g^s is

$$\mathcal{T}_g^s = \mathcal{DF}(\pi_1(F_g^s), \mathrm{PSL}_2(\mathbb{R}))/\mathrm{PSL}_2(\mathbb{R}),$$

where $\mathcal{DF}$ denotes the set of discrete, faithful representations of $\pi_1(F_g^s)$ into $\mathrm{PSL}_2(\mathbb{R})$, and $\mathrm{PSL}_2(\mathbb{R})$ acts by conjugation.

Lemma 2.2. (Aramayona [17]) *The two difinitions are equivalent.*

Proof. A point $[(M, f)] \in \mathcal{T}_g^s$ determines a conjugacy class of faithful representations of $\pi_1(M) \cong \pi_1(F_g^s)$ via the holonomy map. Conversely, given a representation ρ, $M = \mathbb{H}/\rho(\pi_1(F_g^s))$ comes equipped with a hyperbolic structure. Then ρ induces a homotopy equivalence $h : F_g^s \to M$ which is then homotopic to a homeomorphism $f : F_g^s \to M$. Finally, any two conjugate representations produce isometric surfaces. □

We now define a slight generalisation of the Teichmüller space, called the decorated Teichmüller space. Let $\rho \in \mathcal{T}_g^s$ and $\Gamma_m = \rho(\pi_1(F_g^s))$, which is a subgroup of $\mathrm{SO}^+(1, 2)$. There is a corresponding covering map $\mathbb{H} \to F_g^s = \mathbb{H}/\Gamma_m$. We shall also consider the corresponding action on $\mathbb{D}$ with the covering map $\mathbb{D} \to F_g^s$. Note the Poincaré metric on $\mathbb{D}$ projects to a metric on F_g^s with the corresponding geodesics, which we call "Γ-geodesics". Similarly, the projected image of horocycles are called "Γ-horocycles". Represent a point in $\mathcal{T}_g^s$ by $\Gamma_m < \mathrm{SO}^+(1, 2)$, and choose a distinguished Γ-horocycle h_i around each cusp x_i corresponding to the puncture x_i. Let $z_i \in L^+$ be the point corresponding to h_i and set $B_i = \Gamma z_i$. The *decorated Teichmüller space* of F_g^s is

$$\tilde{\mathcal{T}}_g^s = \{(\Gamma_m, B_1, \ldots, B_s) : \Gamma_m \in \mathcal{T}_g^s\}/\mathrm{SO}^+(1, 2).$$

Let MC_g^s denote the full mapping class group of isotopy classes of orientation-preserving homeomorphisms of F_g^s. MC_g^s acts on $\tilde{\mathcal{T}}_g^s$ in the natural way by change of marking.

Let $\tilde{\Gamma}_m = (\Gamma_m, B_1, \ldots, B_s) \in \tilde{\mathcal{T}}_g^s$. Set $\mathcal{B} = B_1 \cup \cdots \cup B_s$ and let C be the closed convex Euclidean hull of $\mathcal{B}$ in $\mathbb{M}$. Then we have

Proposition 2.3. (Penner [70]) *The boundary of C inside L^+ consists of a countable set of codimension 1 "faces", each of which is the convex hull of a finite number of points in $\mathcal{B}$.*

For $\tilde{\Gamma}_m \in \tilde{\mathcal{T}}_g^s$, let $\Delta(\tilde{\Gamma}_m)$ to be the collection of geodesics on F_g^s arising in the following way: Consider, by proposition 2.3, the boudnary ∂C of C, which is the convex hull of $\mathcal{B} = B_1 \cup \cdots \cup B_s$. The edges of ∂C inside L^+ has ends in $\mathcal{B}$. If $z, w \in \mathcal{B}$ are two ends of an edge, the geodesic in $\mathbb{D}$ connecting $\bar{z}, \bar{w} \in S^1_\infty$ projects to a geodesic connecting the corresponding cusps of F_g^s. $\Delta(\tilde{\Gamma}_m)$ is the collection of such geodesics. In fact, $\Delta(\tilde{\Gamma}_m)$ defines a cell decomposition of F_g^s.

Theorem 2.4. (Penner [70]) $\Delta(\tilde{\Gamma}_m)$ *consists of a finite collection of simple geodesic arcs disjointly embedded in F_g^s connecting punctures. Furthermore, components of $F_g^s - \Delta(\tilde{\Gamma}_m)$ are simply connected.*

The isotopy class of such a decomposition is called an *ideal cell decomposition*, or an i.c.d. of F_g^s. Let Δ be an i.c.d. of F_g^s, and define

$$\mathcal{C}_0(\Delta) = \{\tilde{\Gamma}_m \in \tilde{\mathcal{T}}_g^s : \Delta(\tilde{\Gamma}_m) = \Delta\},$$
$$\mathcal{C}(\Delta) = \{\tilde{\Gamma}_m \in \tilde{\mathcal{T}}_g^s : \Delta(\tilde{\Gamma}_m) \subseteq \Delta\}.$$

The result we need is the following;

Theorem 2.5. (Penner [70]) *If Δ is an i.c.d. of F_g^s, then $\mathcal{C}_0(\Delta)$ is an open cell of dimension $\#\Delta$. $\{\mathcal{C}(\Delta) : \Delta$ is an i.c.d. of $F_g^s\}$ is a MC_g^s-invariant cell decomposition of $\tilde{\mathcal{T}}_g^s$ itself.*

The proof is rather long and technical, so we present only a sketch of it. The idea is to first establish the equivalence between the condition $\tilde{\Gamma}_m \in \mathcal{C}_0(\Delta)$ and so-called *face condition* between the "length" of geodesic arcs around each $e \in \Delta$. The "length" $\Lambda(e)$ of an arc $e \in \Delta$ connecting cusps x_i and x_j is defined as follows. Lift e to $\mathbb{D}$ and let $\bar{z}, \bar{w} \in S^1_\infty$ be two points in the lift of x_i and x_j, respectively. Let $z \in B_i$ and $w \in B_j$ be the points that lie on the rays in L^+ corresponding to $\bar{z}$ and $\bar{w}$, respectively, and define

$$\Lambda(e) = \sqrt{-\langle z, w \rangle}.$$

Note $\Lambda(e)$ depends on $\tilde{\Gamma}_m$, it does not, however, depend on the choice of $\bar{z}, \bar{w} \in S^1_\infty$, nor the group Γ_m. To define the face condition, we assume Δ is maximal with respect to inclusion, so that the lengths of the geodesics in Δ provides a coordinate for $\tilde{\mathcal{T}}_g^s$. In particular, a lift e separates two triangles. Label the edges in the two triangles as in figure 9 and let $\pi : \mathbb{D} \to F_g^s$, so that $\pi(\tilde{a}) = a, \ldots, \pi(\tilde{d}) = d$. The *face condition* on $e \in \Delta$ is the inequality

$$\begin{aligned} F(e) =& \Lambda(a)\Lambda(b)\left(\Lambda^2(c) + \Lambda^2(d) - \Lambda^2(e)\right) \\ &+ \Lambda(c)\Lambda(d)\left(\Lambda^2(a) + \Lambda^2(b) - \Lambda^2(e)\right) > 0. \end{aligned}$$

The proof of this equivalence depends on the Λ-length relations in $\mathbb{H}$ and L^+, but we omit the details here. We then claim there is an embedding of $\tilde{\mathcal{T}}_g^s$ in $\mathbb{R}^{\mathcal{E}}$, where $\mathcal{E} = \mathcal{E}(\Delta)$ is the set of ends of the triangles in Δ. To see this, let

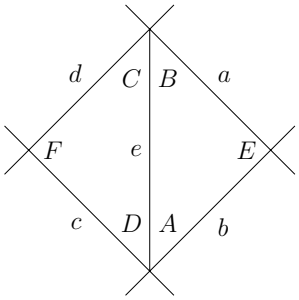


Figure 9: Labelling of edges and ends around a lift of an arc e.

$\tilde{\Gamma}_m \in \tilde{\mathcal{T}}_g^s$ and let $T \subset F$ be a triangle in Δ, with sides $\{c, d, e\}$. Let $\tilde{T}$ be a lift of T in $\mathbb{D}$, and let C, D, E be the ends opposite to lifts of c, d, e, respectively. Define a map

$$I : \tilde{\mathcal{T}}_g^s \approx \mathbb{R}_+^\Delta \to \mathbb{R}_+^\mathcal{E}$$

by

$$h(E, \tilde{\Gamma}_m) = \frac{\Lambda(e)}{\Lambda(c)\Lambda(d)}$$

for each end $E \in \mathcal{E}(\Delta)$. We see immediately

$$\Lambda^{-2}(e) = h(C, \tilde{\Gamma}_m)h(D, \tilde{\Gamma}_m),$$

so I is indeed an embedding. Moreover, if $e \in \mathcal{E}$ "abuts" on ends A, B, C, D as in figure 9, then we have a relation

$$h(A, \tilde{\Gamma}_m)h(B, \tilde{\Gamma}_m) = h(C, \tilde{\Gamma}_m)h(D, \tilde{\Gamma}_m),$$

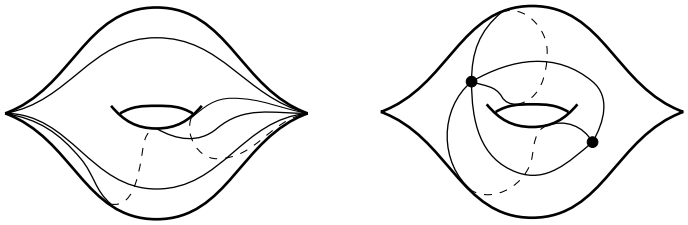
which we call the *coupling equation* of e. It follows that $I(\tilde{\mathcal{T}}_g^s) \subset \mathbb{R}_+^\mathcal{E}$ is characterised by the coupling equations. Now define a pair of vectors $B_e, C_e \in \mathbb{R}^\mathcal{E}$ for each $e \in \Delta$, where B_e and C_e lie in the coordinate subspace of $\mathbb{R}^\mathcal{E}$ corresponding to the ends A, B, C, D around e. Specifically B_e has an entry $(1, 1, 1, 1)$ and C_e has $(1, -1, 1, -1)$. It can be shown that $\{B_e, C_e : e \in \Delta\}$ is a basis for $\mathbb{R}^\mathcal{E}$, and that the face condition depends only on B_e's. More precisely, if $z \in \mathbb{R}^\mathcal{E}$, we may write $z = \sum x_e B_e + \sum y_e C_e$, where the sums run over all $e \in \Delta$. Then z satisfies the face relation on Δ (recall the embedding I of $\mathbb{R}^\Delta$ in $\mathbb{R}^\mathcal{E}$), if and only if $x_e > 0$ for all $e \in \Delta$. Let $X = \{\sum x_e B_e : x_e \in \mathbb{R}\}$, $Y = \{\sum y_e C_e : y_e \in \mathbb{R}\}$, $\bar{X} = \{\sum x_e B_e : x_e \geq 0\}$, $X_0 = \{\sum x_e B_e : x_e > 0\}$ be subspaces of $\mathbb{R}^\mathcal{E}$. Note $\bar{X}$ has a structure as a cone on a simplex, with the faces of $\bar{X}$ corresponding to subsets $\Delta' \subset \Delta$, where $F(e) > 0$ on $e \in \Delta'$ and $F(e) = 0$ on $e \in \Delta - \Delta'$. Call a face F of $\bar{X}$ *finite* if the corresponding subset $\Delta' = \{e : x_e \neq 0\}$ of Δ is an i.c.d., and define

$$X^+ = X_0 \cup \{\text{faces } F \text{ of } \bar{X} : F \text{ is finite}\} \subset \bar{X}.$$

We think of $\tilde{\mathcal{T}}_g^s$ as a subset of $\mathbb{R}_+^\mathcal{E} \subset \mathbb{R}^\mathcal{E}$ determined by the coupling equations, and consider the projection Π of $\mathbb{R}^\mathcal{E}$ along Y onto X. It turns out Π induces a homeomorphism of $\mathcal{C}(\Delta)$ onto X^+ for each maximal i.c.d. Δ. Theorem 2.5 then follows. Note by definition,

$$\mathcal{C}_g^s = \{\mathcal{C}(\Delta) : \Delta \text{ is an i.c.d. of } F_g^s\}$$

is isomorphic to the poset of i.c.d.'s of F_g^s with the relation of inclusion. Furthermore, an i.c.d. of F_g^s corresponds to a spine of F_g^s via the duality described in section 2.1 (figure 10). Hence we obtain



(a) (b)

Figure 10: Duality correspondence between an i.c.d. of F_1^2 (figure 10a) and a fatgraph spine on F_1^2 (figure 10b).

Theorem 2.6. (Penner [69]) *Suppose $2g-2+s>0$. Then there is a $MC(F_g^s)$-invariant cell decomposition of decorated Teichmüller space $\tilde{\mathcal{T}}_g^s$ which is isomorphic to the combinatorial space of all homotopy classes of positive metric fatgraph spines $\tau \subset F$, with each vertex having valence at least three.*

We will need an extension of this result to bordered surfaces $F_{g,r}$ with genus g, r boundary components denoted ∂_i with $i = 1, \ldots, r$, and no punctures. In this setup, univalent vertices correspond to points $d_i \in \partial_i$ for each i. A metric on $F_{g,r}$ does not assign any value to an edge incident on a point on a boundary, and only those edges with metric values can be contracted, still subject to the no-vanishing cycle condition (see section 2.1).

Let $D = \{d_1, \ldots, d_r\}$. A *quasi hyperbolic metric* on $F_{g,r}$ is a hyperbolic metric on $F - D$, such that $\partial_i - \{d_i\}$ is totally geodesic for all i. The decorated Teichmüller space of $F_{g,r}$ is the space $\tilde{T}_{g,r} = \tilde{T}(F_{g,r})$ of all quasi hyperbolic metrics on $F - D$, together with a specification of a segment of a horocycle centred at d_i for each i, modulo push-forward by diffeomorphisms of $F - G$ which are isotopic to the identity. We require that the mapping class group $MC(F_{g,r})$ of homeomorphisms fix the boundary distinguished points setwise and homotopies of homeomorphisms must fix them pointwise. Gluing two copies of $F_{g,r}$ along their boundaries produces a closed surface, to which we can apply the arguments for the punctured surfaces, as described in [68]. Thus we obtain

Theorem 2.7. (Penner [69]) *Suppose $g+r-1>0$. Then there is a $MC(F_{g,r})$-invariant cell decomposition of decorated Teichmüller space $\tilde{T}_{g,r}$ which is homotopy equivalent to the combinatorial space of all isotopy clases of positive metric fatgraph spines $\tau \subset F_{g,r}$. The univalent vertices of τ lie in the boundary, with exactly one in each boundary component, and the remaining vertices have valence at least three.*

We end this section by noting that the quotient of Teichmüller space $\mathcal{T}(F)$ of possibly punctured or bordered surface by the mapping class group is the Riemann moduli space

$$M(F) = \mathcal{T}(F)/MC(F).$$

From theorems 2.6 and 2.7, we have cell decompositions of the space

$$\tilde{M}(F) = \tilde{\mathcal{T}}(F)/MC(F).$$

Since $M(F)$ and $\tilde{M}(F)$ are homotopy equivalent, the Penner-Strebel cell decomposition can be considered as the cell decomposition of $M(F)$.

2.3 Matrix model

A matrix model is characterised by a space of matrices E (also called a matrix ensemble), and a measure $\mathrm{d}M$ on E. Given a function (also called the potential) V on E, we define the partition function or matrix integral by

$$Z = \int_E e^{-N\mathrm{Tr}V(M)}\,\mathrm{d}M. \tag{3}$$

V must satisfy some requirements for the integral to be well-defined, but that will not be an issue for us, as we will see later. There are three matrix ensembles called the Gaussian ensembles, which are:

- the ensemble of real symmetric matrices, called the Gaussian Orthogonal Ensemble (GOE),
- the ensemble of complex Hermitian matrices, called the Gaussian Unitary Ensemble (GUE),
- and the ensemble of quaternionic Hermitian matrices, called the Gaussian Symplectic Ensemble (GSE).

We denote these Gaussian ensembles by E_N^β, where $\beta \in \{1, 2, 4\}$, depending on whether it is GOE ($\beta = 1$), GUE ($\beta = 2$), or GSE ($\beta = 4$). Each of the ensembles can be realised as a space of $N \times N$ matrices, whose coefficients are real if $\beta = 1$, complex if $\beta = 2$, and quaternionic if $\beta = 4$.

A matrix M in a Gaussian ensemble has real eigenvalues, and can be diagonalised as

$$M = U\Lambda U^{-1}, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N),$$

where U belongs to $O(N)$ if $M \in E_N^1$, $U(N)$ if $M \in E_N^2$, and $Sp(2N)$ if $M \in E_N^4$.

Each of the above Gaussian ensembles E_N^β has a Lebesgue measure given by

$$\mathrm{d}M = \prod_{i=1}^{N} \mathrm{d}M_{i,i} \prod_{i<j} \prod_{\alpha=0}^{\beta-1} \mathrm{d}M_{i,j}^{(\alpha)},$$

where $M_{i,j}^{(\alpha)}$ denotes the real and imaginary components of $M_{i,j}$. If, for example $\beta = 2$, we have

$$\mathrm{d}M = \prod_{i=1}^{N} \mathrm{d}M_{i,i} \prod_{i<j} \mathrm{dRe}M_{i,j}\,\mathrm{dIm}M_{i,j}.$$

Given a potential V, we can obtain another measure $\mathrm{d}\mu$ by

$$\mathrm{d}\mu = e^{-N\mathrm{Tr}V(M)}\,\mathrm{d}M.$$

Surprisingly the problem of enumerating orientable fatgraphs turns out to be solvable by considering the integrals of certain functions over GUE, the ensemble of Hermitian matrices ([1, 25]). To illustrate this remarkable connection, let us start with a toy example given in [64]. A fundamental result for this section and beyond is Wick's theorem, which states the expectation value of a product of Gaussian random variables can be computed as the sum over all pairings of product of expectation values of pairs;

Theorem 2.8. (Wick; see Bessis et al. [21]) *Let $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and let A be a symmetric, positive definite $n \times n$ matrix. Consider the integral*

$$\langle x_{i_1} x_{i_2} \cdots x_{i_k} \rangle = \frac{\int x_{i_1} x_{i_2} \cdots x_{i_k} \exp\left(-\frac{1}{2} \sum_{\mu,\nu} x_\mu A_{\mu\nu} x_\nu\right) \mathrm{d}x}{\int \exp\left(-\frac{1}{2} \sum_{\mu,\nu} x_\mu A_{\mu\nu} x_\nu\right) \mathrm{d}x}.$$

Then we have

$$\begin{aligned} \langle x_{i_1} x_{i_2} \cdots x_{i_{2k+1}} \rangle &= 0, \\ \langle x_{i_1} x_{i_2} \rangle &= \left(A^{-1}\right)_{i_1 i_2}, \\ \langle x_{i_1} x_{i_2} \cdots x_{i_{2k}} \rangle &= \sum_{\text{parings}} \prod_{(s,t)} \langle x_{i_s} x_{i_t} \rangle, \end{aligned}$$

where the sum is over all pairings of the indices $i_1, \ldots, i_{2k}$.

For the toy example, let $x, y \in \mathbb{R}^n$ and (x, y) be the Euclidean inner product in $\mathbb{R}^n$. For a function $f(x)$ on $\mathbb{R}^n$, set

$$\langle f \rangle = \frac{\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(x,x)\right) f(x) \, \mathrm{d}x}{\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(x,x)\right) \mathrm{d}x}.$$

Then we see that

$$\langle x_i x_j \rangle = \delta_{ij}.$$

So Wick's theorem implies, for example,

$$\langle x_i x_j x_k x_l \rangle = \langle x_i x_j \rangle \langle x_k x_l \rangle + \langle x_i x_k \rangle \langle x_j x_l \rangle + \langle x_i x_l \rangle \langle x_j x_k \rangle = \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}.$$

The different pairings of indices i, j, k, l can be represented as the different ways of pairing the four indexed half-edges of a four-valent vertex (figure 11).

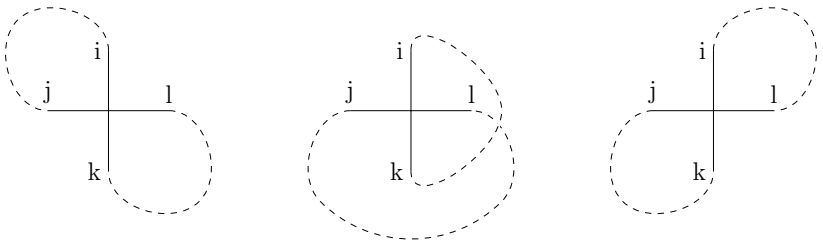


Figure 11: Three different pairings of indices i, j, k, l.

Now consider E_N^2, the ensemble of $N \times N$ Hermitian matrices and let f be a conjugation-invariant function on it. We will consider the expectation of $f(M)$,

$$\langle f(M) \rangle = \frac{\int_{E_N^2} \exp(-\frac{1}{2}\mathrm{Tr}M^2) f(M)\, \mathrm{d}M}{\int_{E_N^2} \exp(-\frac{1}{2}\mathrm{Tr}M^2)\, \mathrm{d}M}.$$

Since f is conjugation-invariant, and the measure $\mathrm{d}M$ is invariant under changes of bases, this is just a Gaussian integral, and we may apply Wick's theorem. Since

$$\mathrm{Tr}M^2 = \sum_{i,j} M_{ij} M_{ji}, \tag{4}$$

we deduce that the so-called *Wick contraction* is given by

$$\langle M_{ij} M_{kl} \rangle = \delta_{il}\delta_{kj}.$$

This can be seen by writing the matrix M in vector form, and expressing the quadratic form $\mathrm{Tr}M^2$ as a $N^2 \times N^2$ matrix. This matrix A has the form

$$A_{N(i-1)+j,N(j-1)+i} = 1, \quad 1 \leq i, j \leq N$$

and 0 everywhere else. In particular, $A^{-1} = A$ and the index that corresponds to $M_{ij}M_{kl}$ is $(N(i-1)+j, N(k-1)+l)$. Thus by Wick's theorem, we obtain (4).

Again using Wick's theorem, we compute

$$\begin{aligned}
\left\langle \mathrm{Tr}M^4 \right\rangle &= \left\langle \sum_{i,j,k,l} M_{ij}M_{jk}M_{kl}M_{li} \right\rangle \\
&= \sum_{i,j,k,l} \langle M_{ij}M_{jk}M_{kl}M_{li} \rangle \\
&= \sum_{i,j,k,l} \left(\langle M_{ij}M_{jk} \rangle \langle M_{kl}M_{li} \rangle + \langle M_{ij}M_{kl} \rangle \langle M_{jk}M_{li} \rangle + \langle M_{ij}M_{li} \rangle \langle M_{jk}M_{kl} \rangle \right) \\
&= \sum_{i,j,k,l} \left(\delta_{ik}\delta_{jj}\delta_{ki}\delta_{ll} + \delta_{il}\delta_{kj}\delta_{ji}\delta_{lk} + \delta_{ii}\delta_{jl}\delta_{jl}\delta_{kk} \right) \qquad (5) \\
&= \left(2N^3 + N \right). \qquad (6)
\end{aligned}$$

For the graphical representation of the different pairings of entries in the matrix M, we use the "fattened" version of the four-valent vertex we used above, with the half-edges now shown as double lines labelled by double indices. Each half-edge has one line oriented away from, and one oriented towards the vertex. We also require that the orientation of the half-edges are consistent, so that the inward-pointing edge i is connected to the outward pointing edge of the same label (figure 12). Now we can interpret the first summand in equation (5) as connecting i-out with k-in, j-out with j-in, k-out with i-in, and l-out with l-in. This results in the first surface in figure 13. Similarly the second and third summands correspond to the second and third surfaces in figure 13. Note in equation (6), the powers of N correspond to the number of boundary cycles in the associated surfaces, and the integral coefficients of N^j is the number of ways of obtaining the same graph by gluing the available half-edges, i.e. #Glu. We can in general obtain the number of fatgraphs with one m-valent vertex in this manner.

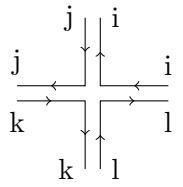


Figure 12: "Fattened" four-valent vertex with the oriented double half-edges.

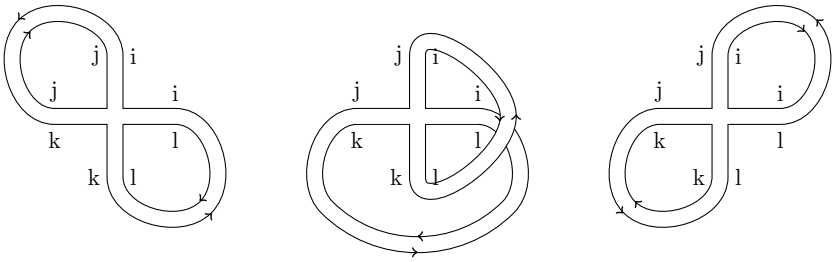


Figure 13: Three different surfaces obtained by connecting the half-edges of the fattened four-valent vertex in figure 12.

Lemma 2.9. *Let $M \in E_N^2$, the ensemble of $N \times N$ Hermitian matrices, and $j \in \mathbb{N}$. Then we have*

$$\left\langle \mathrm{Tr} M^j \right\rangle = \sum_\tau \#\mathrm{Glu}(\tau) N^{n(\tau)},$$

where $n(\tau)$ is the number of boundary cycles of fatgraph τ, and the sum is over all topologically distinct fatgraphs with one j-valent vertex and no unconnected half-edges.

Wick's theorem further implies that products of traces correspond to multiple vertices, valencies of which are determined by the powers of M. In other words,

$$\left\langle \prod_{k=1}^m \mathrm{Tr} M^{j_k} \right\rangle = \sum_\tau \#\mathrm{Glu}(\tau) N^{n(\tau)}, \tag{7}$$

where the sum is now over all fatgraphs with m vertices of valencies $j_1, j_2, \ldots, j_m$, which may not be connected.

Let us consider another integral by setting

$$V(M) = \frac{N}{2} M^2$$

and

$$\langle f \rangle = \frac{1}{Z_0} \int e^{-N \mathrm{Tr} \frac{M^2}{2}} f(M) \, \mathrm{d}M,$$

where Z_0 is the normalisation constant, given in this case by

$$Z_0 = \int_{E_N^2} \mathrm{d}M e^{-N \mathrm{Tr} \frac{M^2}{2}} = 2^N \left(\frac{\pi}{N} \right)^{\frac{N^2}{2}}.$$

Then we have

$$\langle M_{ij}M_{kl}\rangle = \frac{1}{N}\delta_{il}\delta_{jk}.$$

Suppose we have k_j j-valent vertices, $j = 1, \ldots, m$. Using lemma 2.1, we can express (7) in another way;

$$\left\langle \prod_{j=1}^{m} \frac{1}{k_j!}(\frac{tN}{j}\mathrm{Tr}M^j)^{k_j} \right\rangle = \sum_{\tau} \frac{1}{\#\mathrm{Aut}(\tau)} N^{\chi(\tau)} t^{\sum_j k_j}, \tag{8}$$

where $\chi(\tau)$ is the Euler characteristic, and the sum is now over all fatgraphs with m vertices, m_j of which are j-valent. For the power of N, we have a negative contribution from $1/N$ in the Wick contraction counting the number of edges. By lemma 2.9, we have a positive contribution from the number of boundary components, and from tN in equation (8), which counts the number of vertices. In other words, the power of N is given by

$$\#\text{vertices} - \#\text{edges} + \#\text{boundaries} = \chi(\tau).$$

Let

$$V(M) = \frac{M^2}{2} - \sum_{j=3}^{\infty} \frac{t_j}{j} M^j \tag{9}$$

and define a formal matrix integral by setting

$$\frac{1}{Z_0}\int_{\text{formal}} \mathrm{d}M e^{-N\mathrm{Tr}V(M)} = \sum_{\tau} \frac{1}{\#\mathrm{Aut}(\tau)} N^{\chi(\tau)} t_j^{k_j}, \tag{10}$$

where the sum is over all fatgraphs, with any number of vertices, and any combinations of valencies. It is important to note that this is a formal integral, with no implication that the integral is convergent. However, the definition is a natural one, if we think of it as expanding the exponential of the non-quadratic part of $V(M)$ and then exchanging the integral and the summation;

$$\begin{aligned}
\frac{1}{Z_0}\int_{\text{formal}} \mathrm{d}M e^{-N\mathrm{Tr}V(M)} &= \frac{1}{Z_0}\int_{\text{formal}} \mathrm{d}M e^{-N\mathrm{Tr}\frac{M^2}{2}} e^{N\sum_{j=3}^{\infty}\frac{t_j}{j}\mathrm{Tr}M^j} \\
&\text{“}=\text{”} \left\langle \sum_{k=0}^{\infty}\frac{1}{k!}\left(\sum_{j=3}^{\infty}\frac{Nt_j}{j}\mathrm{Tr}M^j\right)^k \right\rangle \\
&\text{“}=\text{”} \sum_{k=0}^{\infty}\frac{1}{k!}\sum_{j_1,\ldots,j_k}\left\langle \prod_{l=1}^{k}\frac{Nt_{j_l}}{j_l}\mathrm{Tr}M^{j_l}\right\rangle,
\end{aligned}$$

where the second sum is over all values of j_l, $l = 1, \ldots, k$. This sum contains permutations of j_l's that do not change the number of j_l's with the same value, which correspond to the relabelling of vertices of the same valencies. The number of such permutations is given by $\frac{k!}{\prod k_j!}$, which cancels the $k!$ in the denominator and replaces it with $\prod \frac{1}{k_j!}$ in equation (8). Thus we see that the definition of the formal integral (10) is indeed a natural one.

Up to now we have only considered GUE, E_N^2 in our formal integral. Similar constructions are possible for the two other ensembles, E_N^1 and E_N^4. In general,

as discussed in [38], for $\beta \in \{1, 2, 4\}$, with the potential

$$V(M) = \frac{M^2\beta}{4} - \sum_{j=3}^{\infty} \frac{\beta t_j}{2j} M^j,$$

we get the Wick contraction

$$\langle M_{ij} M_{kl} \rangle = \epsilon_1 \delta_{il}\delta_{kj} - (\epsilon_1 + \epsilon_2)\delta_{ik}\delta_{lj},$$

where

$$\epsilon_1 = \frac{1}{N}, \quad \epsilon_2 = -\frac{2}{N\beta}.$$

Graphically, we can represent the second term in the propagator $\delta_{ik}\delta_{jl}$ by a twisted band. This allows us to extend the formal integral to count the number of non-orientable graphs. We see that E_N^2 is in fact a special case where $\epsilon_1 + \epsilon_2$ vanishes.

2.4 Topological recursion

In this section we describe topological recursion, a framework developed by Chekhov, Eynard and Orantin [28, 43]. Topological recursion provides a recursive formula for all terms in the large N expansion of the formal matrix integral. The central object in topological recursion is a so-called spectral curve, which arises from the leading order loop equation of a matrix model. The spectral curve can be characterised as a curve on which the resolvent, which is related to the density of eigenvalues of the random matrix under consideration. An important aspect of topological recursion is that it does not depend on the underlying matrix model, but on the spectral curve. In other words, topological recursion can be utilised beyond matrix models (see [43] or [10] for some examples). Here we start by reviewing the formal setup of topological recursion, following the discussion given in [11]. We then describe different objects in the formal setup in more details in the context of matrix models, following the discussion in [12].

2.4.1 Formal setup

Let $U = \sqcup_i U_i$ be a Riemann surface, where each U_i is a neighbourhood of a point o_i, and let $x : U \to V$ be a branched covering branched at $O = \sqcup_i \{o_i\}$ with $o_i \in U_i$ for each i. If we assume that x has only simple ramifications, then we can choose a coordinate z_i on each U_i, such that $x(z_i) = (z_i)^2/2 + x(o_i)$. Furthermore, let σ_i be a holomorphic involution on U_i, which sends z_i to $-z_i$. Let $\Delta = \cup_i \{(z_i, z_i) \in U^2 \mid z_i \in U_i\}$ be the diagonal of $U \times U$ and K_U be the canonical bundle of U.

Before describing the topological recursion, let us give the following definition.

Definition 3. Let $E_1 \to X_1$ and $E_2 \to X_2$ be two vector bundles over manifolds X_1 and X_2. The product manifold $X_1 \times X_2$ has two natural pullback bundles, $\pi_1^* E_1 \to X_1 \times X_2$ and $\pi_2^* E_2 \to X_1 \times X_2$, where π_1 and π_2 are the projection onto the first and second factors, respectively. The external tensor product $E_1 \boxtimes E_2$ is given by

$$E_1 \boxtimes E_2 := \pi_1^* E_1 \otimes \pi_2^* E_2. \tag{11}$$

Note $E_1 \boxtimes E_2$ is a bundle over $X_1 \times X_2$. If $E_1 = E_2 = E$, we write $E^{\boxtimes 2} := E \boxtimes E$.

The initial data of topological recursion is as follows. To simplify the notation, we omit the index i in z_i where there is no ambiguity.

- A holomorphic 1-form $\omega_{0,1} \in H^0(U, K_U)$, such that $\omega_{0,1}(z) - \omega_{0,1}(\sigma(z)) = 2\omega_{0,1}(z)$ has at most double zeros at o_i.
- A symmetric, holomorphic form $\omega_{0,2} \in H^0(U^2, K_U^{\boxtimes 2}(2\Delta))^{\mathfrak{S}_2}$, where $K_U^{\boxtimes 2}$ denotes the tensor product of pullback bundles as defined above, and 2Δ denotes that $\omega_{0,2}$ has double poles along Δ. $\mathfrak{S}_2$ denotes the invariance under the action of the symmetric group $\mathfrak{S}_2$ on the U factors.

In local coordinates, we can write the form $\omega_{0,1}$ as

$$\omega_{0,1}(z) = \sum_{d \geq 0} \omega_{0,1}^{(i;d)} z^d \, \mathrm{d}z \tag{12}$$

for $z \in U_i$, with the extra condition $\omega_{0,1}^{(i;0)} \neq 0$ or $\omega_{0,1}^{(i;2)} \neq 0$, corresponding to the condition that $\omega_{0,1}(z) - \omega_{0,1}(\sigma(z))$ has at most double zeros in O. The form $\omega_{0,2}$ can be written as

$$\omega_{0,2}(z_{i_1}, z_{i_2}) = \frac{\delta_{i_1 i_2} t_{i_1} \, \mathrm{d}z_{i_1} \, \mathrm{d}z_{i_2}}{(z_{i_1} - z_{i_2})^2} + \sum_{d_1, d_2 \geq 0} \omega_{0,2}^{(i_1, i_2; d_1, d_2)} z_{i_1}^{d_1} z_{i_2}^{d_2} \, \mathrm{d}z_{i_1} \, \mathrm{d}z_{i_2}, \tag{13}$$

for $z_{i_1} \in U_{i_1}$ and $z_{i_2} \in U_{i_2}$. It is usually assumed $t_i = 1$ for all i. Using these two forms, we define the recursion kernel

$$\mathcal{K}_i(y, z) = \frac{1}{2} \frac{\int_{\sigma_i(z)}^{z} \omega_{0,2}(\cdot, y)}{\omega_{0,1}(z) - \omega_{0,1}(\sigma_i(z))}. \tag{14}$$

If we let $\Delta_\sigma = \sqcup_i \{(z, \sigma_i(z)) \mid z \in U_i\}$, we see that $\mathcal{K}_i$ has poles in $\Delta \sqcup \Delta_\sigma$, since $\omega_{0,2}$ has poles in Δ and $\omega_{0,1}(z) - \omega_{0,1}(\sigma_i(z))$ has zeros in Δ_σ. We also see $\mathcal{K}_i$ has the form $\varphi_i(y, z) \, \mathrm{d}y \otimes \frac{\partial}{\partial z}$ for some function φ_i. In other words, we have

$$\mathcal{K}_i \in H^0(U \times U_i, [K_U \boxtimes K_{U_i}^{-1}(O)](\Delta \sqcup \Delta_\sigma)). \tag{15}$$

The topological recursion provides a sequence of symmetric n-forms

$$\omega_{g,n} \in H^0(U^n, (K_U(\star O))^{\boxtimes n})^{\mathfrak{S}_n} \tag{16}$$

for $2g - 2 + n > 0$. Let $I = (z_{i_2}, \ldots, z_{i_n})$. The recursion formula for topological recursion is

$$\begin{aligned} \omega_{g,n}(z_{i_1}, I) = \sum_{o_i \in O} \operatorname*{Res}_{z \to o_i} \mathcal{K}_i(z_{i_1}, z) \Bigg\{ & \omega_{g-1,n+1}(z, \sigma_i(z), I) \\ & + \sum_{(h,J)} \omega_{h,1+|J|}(z, J) \otimes \omega_{h',1+|J'|}(\sigma_i(z), J') \Bigg\}, \end{aligned} \tag{17}$$

where the second sum is over all non-trivial partitions of the pair (g, I); i.e. the pairs (h, J) and (h', J'), such that $h + h' = g$ and $J \sqcup J' = I$, with the condition $(h, J) \neq (0, \emptyset)$ and $(h, J) \neq (g, I)$. We observe that $\omega_{g,n}$ is symmetric in the

factors of I by construction. However the symmetry in all factors, in particular between z_{i_1} and z_{i_2}, is not trivial. See, for example, [10] for a proof of full symmetry. We also obtain a sequence of numbers $(F_g)_{g\geq 2}$, given by

$$F_g = \frac{1}{2-2g} \sum_{o_i \in O} \operatorname*{Res}_{z \to o_i} \left(\oint_{o_i}^{z} \omega_{0,1} \right) \omega_{g,1}(z). \tag{18}$$

2.4.2 Topological recursion and matrix models

We will now apply the topological recursion framework to matrix models, following the presentation in [12].

Let us now consider a general Hermitian matrix model,

$$Z = \int \mathrm{d}M e^{-\frac{1}{\hbar}\mathrm{Tr}V(M)}, \tag{19}$$

where we have replaced N in equation (3) by $\frac{1}{\hbar}$. The two are related by the so-called 't Hooft coupling

$$T = \hbar N = \text{const},$$

which is kept fixed. For the purpose of fatgraph enumeration, we can simply set $T = 1$.

We start by showing how to find the spectral curve in the matrix model setting. The spectral curve is related to the resolvent, which is the leading order term in the expansion of the all-order resolvent, defined as

$$\omega_1(x) = \hbar \left\langle \mathrm{Tr} \frac{1}{x - M} \right\rangle = \sum_{g=0}^{\infty} \hbar^{2g} \omega_1^{(g)}(x). \tag{20}$$

The resolvent $\omega_1^{(0)}$ is also related to the density of eigenvalues $\rho(x)$, which becomes continuous in large N limit, with the support $\mathrm{supp}\rho$ being compact intervals, also called cuts. We have that

$$\omega_1^{(0)}(x) = \frac{1}{t_0} \int_{\mathrm{supp}\rho} \frac{\rho(x')}{x - x'} \, \mathrm{d}x', \tag{21}$$

where

$$t_0 = \int_{\mathrm{supp}\rho} \rho(x) \, \mathrm{d}x.$$

This implies that for large x, $\omega_1^{(0)}$ behaves as

$$\omega_1^{(0)}(x) \underset{x \to \infty}{\sim} \frac{1}{x}.$$

Furthermore, the resolvent satisfies

$$\omega_1^{(0)}(x - i\epsilon) + \omega_1^{(0)}(x + i\epsilon) = \frac{V'(x)}{T}.$$

The resolvent can be determined from these conditions, or by the Migdal formula, which, if we assume that there is a single cut with endpoints a and b, is given by

$$\omega_1^{(0)}(x) = \frac{1}{2T} \oint_{\mathcal{C}} \frac{\mathrm{d}z}{2\pi i} \frac{V'(z)}{x - z} \frac{\sqrt{(x-a)(x-b)}}{\sqrt{(z-a)(z-b)}}, \tag{22}$$

where the contour $\mathcal{C}$ surrounds the cut. We now define a new variable

$$y = \frac{1}{2}V'(x) - T\omega_1^{(0)}(x) = \frac{T}{2}\left(\omega_1^{(0)}(x - i\epsilon) - \omega_1^{(0)}(x + i\epsilon)\right). \tag{23}$$

This variable y is related to x by a polynomial equation, which is the equation that defines the spectral curve. We denote it by

$$A(x, y) = 0. \tag{24}$$

In the context of the topological recursion, a parametric form of this equation, $A(x(p), y(p)) = 0$ is required.

We now move onto defining the two initial data of topological recursion, $W_1^{(0)}$ and $W_2^{(0)}$, corresponding to $\omega_{0,1}$ and $\omega_{0,2}$ in section 2.4.1. These are defined by the correlators

$$\left\langle \mathrm{Tr}\left(\frac{1}{x(p_1) - M}\right) \cdots \mathrm{Tr}\left(\frac{1}{x(p_n) - M}\right) \right\rangle_{\mathrm{conn}} = \sum_{g=0}^{\infty} \hbar^{2g-2+n} \frac{W_n^{(g)}(p_1, \ldots, p_n)}{\mathrm{d}x(p_1) \ldots \mathrm{d}x(p_n)}, \tag{25}$$

where $\langle \cdot \rangle_{\mathrm{conn}}$ denotes the part of the correlator arising from connected diagrams, computed as a formal integral (see section 2.3) with the formal power series

$$\mathrm{Tr}\frac{1}{x - M} = \sum_{k \geq 0} \mathrm{Tr}\frac{M^k}{x^{k+1}}. \tag{26}$$

Equation (25) generalises the all-order resolvent in equation (20). In particular, the one-point correlator is given by the 1-form $W_1^{(g)}(x) = \omega_1^{(g)}(x)\,\mathrm{d}x$.

The 2-form is given by the so-called Bergman kernel $B(p, q)$, which for the one-cut solution takes the form

$$B(p, q) = \frac{\mathrm{d}p\,\mathrm{d}q}{(p - q)^2}. \tag{27}$$

We now have the two forms, given by

$$W_1^{(0)}(p) = 0, \qquad W_2^{(0)}(p_1, p_2) = B(p_1, p_2). \tag{28}$$

The recursion kernel has the form;

$$K(q, p) = \frac{1}{2}\frac{\int_{\sigma q}^{q} B(\cdot, p)}{(y(q) - y(\sigma(q)))\,\mathrm{d}x(p)}.$$

With the above components, we can now use the recursion formula (equation (17)) to compute the multi-point correlators;

$$W_n^{(g)}(p, I) = \sum_{q_i^*} \operatorname*{Res}_{q \to q_i^*} K(q, p) \Bigg(W_{n+1}^{(g-1)}(q, \sigma(q), I) \tag{29}$$

$$+ \sum_{m=0}^{g} \sum_{J \subset I} W_{1+|J|}^{(m)}(q, J) W_{1+n-|J|}^{(g-m)}(\sigma(q), I \setminus J) \Bigg), \tag{30}$$

where $I = (p_1, \ldots, p_n)$.

3 Topology of RNA

RNA is a macromolecule essential in many biological processes, which, apart from carrying information from DNA for protein synthesis, include protein synthesis itself, production of RNA, and regulation of gene expression. RNA molecules that carries the coding sequence for protein production are called coding RNAs, and it is naturally their coding sequence, that is of the primary interest. The RNA molecules that perform other functions are called non-coding RNAs, first of which were discovered in the early 1980s [6, 26].

RNA is a polymer consisting of nucleotides, or bases, together with ribose and phosphate that link nucleotides. The nucleotides in an RNA molecule can form bonds with each other, resulting in the molecule folding. It is widely accepted that the function of a non-coding RNA is deeply related to its structure [47, 85]. In particular, a class of structure called pseudoknots are known to be functionally important in different non-coding RNAs [59, 86]. A pseudoknot is a structure containing bonds between nucleotides that cross each other, and it turns out that fatgraphs provide a framework particularly suited to describing pseudoknot structures [8, 7, 23, 92].

One of the first studies to utilise fatgraphs for investigation of RNA structure was [23], where the authors took RNA structures from available databases and studied their genera. They found that the genus remains small even for large RNA structures, compared to the genus of a randomly generated structure from the same number of nucleotides. Genus was also used to study pseudoknot structures generated by a model [94]. Later Andersen et al. [8] computed generating functions for certain classes of RNA structures by using the fatgraph model of RNA structures. In [77], the genus is used to obtain a (multiple context free) grammar for decomposing pseudoknot structures. The grammar is extended to RNA-RNA interaction structures (i.e. two backbones) in [7]. The same ideas (i.e. decomposition into irreducible structures and topological characteristics of RNA structures) are implemented in an algorithm in [92]. Fatgraphs' relation to matrix model theory (see section 2.3) was used to study enumeration problem for RNA structures [67, 93, 3, 8]. In [12] and [13] Andersen et al. used topological recursion framework to obtain the solution to the matrix model for the structures, where there are no unpaired bases. In order to extend the solutions to RNA structures with unpaired bases, extra combinatorial parameters were introduced in [15, 14].

In what follows, we present a brief overview of these studies, as examples of successful attempts to utilise fatgraphs as a framework for studying biological structures. They will serve as an inspiration to our study of proteins using fatgraphs as a framework. We will start by presenting a brief description of molecular structures of RNA, and how they can be modelled using fatgraphs. We will then present recursion relations for RNA structures in section 3.2, with an emphasis on cut and join methodology. In section 3.3, we present a matrix model for RNA structures. As described in section 2.3, it is a generating function for the number of distinct structures, and these numbers satisfy recursion relations given in section 3.2.

3.1 RNA model

An RNA molecule is a linear sequence of nucleotides, which consist of the following;

- A nucleic acid residue, or "base"
- A ribose sugar containing five carbon atoms
- A phosphate group

The nucleic acid is one of the four compounds; Adenine, Guanine, Uracil and Cytosine, represented by letters A, G, U, and C. It is bonded to the 1' carbon atom in the ribose sugar by a covalent bond. The 5' carbon in the ribose sugar is bonded to the previous phosphate and the 3' carbon to the subsequent phosphate, forming the *backbone* (figure 14). The RNA *primary structure* is the

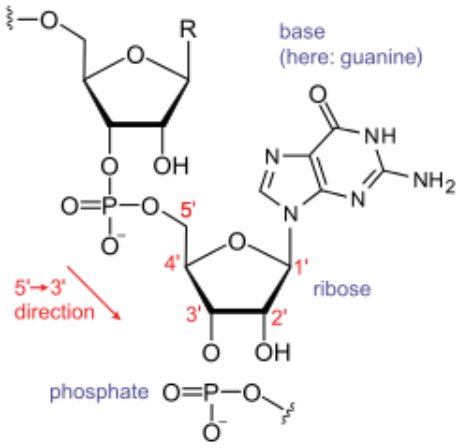


Figure 14: Detailed structure of an RNA. Taken from Wikimedia, W. Sahib, *"Diagram to illustrate 5' to 3' directionality in a nucleic acid"*.

sequence of the bases, and it can be represented as a word in an alphabet with four letters. The acid residues may participate in hydrogen bond according to the Watson-Crick rules, which allow pairings A-U, G-C, and U-G. The set of base pairings, together with the primary structure, is called the RNA *secondary structure*. An RNA molecule can be represented by a diagram, where the backbone is drawn as a horizontal line, and each pairing is drawn as an arc, called a *chord*, above the horizontal line between the paired residues (figure 15). Note we have a natural orientation along the backbone induced by the orientation from 5' end to 3' end, and we will typically draw a backbone with the 5' end to the left. Furthermore, we will only allow at most one chord at any given letter along the backbone. A representation of RNA molecule in this manner is called a *partial linear chord diagram* or simply a *partial chord diagram*. A partial chord diagram is called a *linear chord diagram* or a *chord diagram*, if it contains no unpaired vertex. Since no multiple chords are allowed at any given vertex in our model, the number of vertices in a (linear) chord diagram is necessarily even. A chord between two consecutive letters along the backbone

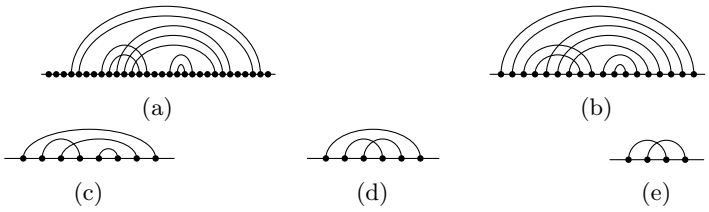


Figure 15: Examples of (a) partial chord diagram, (b) chord diagram, (c) seed, (d) shape, and (e) shadow.

is called a *1-chord*, and a chord between the first and the last vertices in the backbone is called a *rainbow*. Let $\{1, 2, \ldots, n\}$ be the numbering of letters in the backbone from left to right (according to the orientation induced by 5' to 3' orientation). Then each chord may be expressed as a pair of indices (i, j) with $i < j$, showing the positions of start- and end points along the backbone. Two chords (i_1, i_2) and (j_1, j_2) are called *consecutively parallel*, if $i_1 = j_1 - 1$ and $i_2 = j_2 + 1$. (i_1, i_2) and (j_1, j_2) are called *parallel*, if there exists a sequence of consecutively parallel chords linking (i_1, i_2) with (j_1, j_2). More precisely, if there exists an $n \in \mathbb{N}$ and a sequence $\gamma_0 = (i_1, i_2), \gamma_1, \ldots, \gamma_{n-1}, \gamma_n = (j_1, j_2)$ of chords, such that γ_{k-1} and γ_k are consecutively parallel for $k \in \{1, \ldots, n\}$. Being parallel is clearly an equivalent relation, and an equivalence class of parallel chords is called a *stack*. A chord diagram where every stack contains at most one chord is called a *seed*. A seed is called a *shape* if it contains the rainbow and if it contains no 1-chord. A seed is called a *shadow* if it contains no non-crossing arc. A shadow S is *irreducible*, if for any two chords α and β in S, there is a sequence $\gamma_0 = \alpha, \gamma_1, \ldots, \gamma_{n-1}, \gamma_n = \beta$ of chords such that each pair (γ_{i-1}, γ_i) cross each other.

Given a (partial) chord diagram, there is a natural way to equip it with a fatgraph structure, namely by defining an ordering of edges incident at each vertex. This allows us to talk about the genus of a given (partial) chord diagram. Let τ be a shape with one backbone. Then the backbone of τ may be collapsed to a single vertex without altering the number of boundary components, the number of chords, or the Euler characteristic. Hence we have

$$2 - 2g - n = 1 - k,$$

where n is the number of boundary components in τ, and k is the number of chords. We may further take the dual of this collapsed graph (see section 2.1), to obtain a graph with one univalent vertex, and the other vertices having valence at least three (figure 16). Here we recall theorem 2.7, and emphasise the remarkable link between the (dual) fatgraph of RNA shapes of genus g (e.g. figure 16c) and the Penner-Strebel cell decomposition of the Riemann moduli space for a surface $M(F_{g,b})$ of genus g with one boundary component.

An important problem is the enumeration of RNA structures for a given number of chords. We start the discussion with the case of chord diagrams on one backbone. Let $\mathcal{C}_g(k)$ be the set of all chord diagrams with genus g and k chords (i.e. $2k$ vertices), and let $\boldsymbol{c}_g(k)$ be the cardinality of this set, and write $\boldsymbol{C}_g(z) = \sum_{k \geq 0} \boldsymbol{c}_g(k) z^k$ for the generating function. Looking at the graph where the backbone is collapsed to a single vertex, we see that the enumeration of such

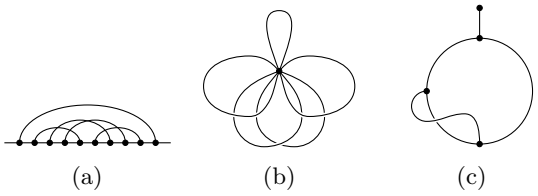


Figure 16: An example of correspondence between a shape and a fatgraph with one univalent vertex. The backbone of a shape (a) is collapsed to obtain a fatgraph with one vertex (b), where the single non-crossing edge corresponds to the rainbow. Its dual (c) is the desired fatgraph with one univalent vertex and all other vertices at least three-valent.

chord diagrams is equivalent to finding the number of ways one can glue the sides of $2n$-gon to produce a closed, orientable surface. Surprisingly the solution to this problem was found by Harer and Zagier in their work on computing Euler characteristics of the moduli space of curves [45]. We present their result here;

Theorem 3.1. (Harer and Zagier [45]) $\boldsymbol{c}_g(k)$ *satisfies the following recursion;*

$$(k+1)\boldsymbol{c}_g(k) = 2(2k-1)\boldsymbol{c}_g(k-1) + (2k-1)(k-1)(2k-3)\boldsymbol{c}_{g-1}(k-2).$$

It follows, for $g \geq 1$*, we have*

$$\mathbf{C}_g(z) = P_g(z)\frac{\sqrt{1-4z}}{(1-4z)^{3g}},$$

where $P_g(z) = \sum_j p_g^{(j)} z^j$ *is a polynomial with integral coefficients, and with degree at most* $(3g-1)$*. Furthermore, we have* $P_g(1/4) \neq 0$*, the coefficient of* z^{2g} *is non-zero and the coefficient of* z^h *is zero for* $0 \leq h \leq 2g-1$*.*

Similarly, let $\mathcal{S}_g(k)$ and $\mathcal{T}_g(k)$ be respectively the collections of all seeds and shapes with genus g and k chords (i.e. $2k$ vertices), and let $\boldsymbol{s}_g(k)$ and $\boldsymbol{t}_g(k)$ be the cardinalities of these sets. Write $\boldsymbol{S}_g(z) = \sum_{k\geq 0} \boldsymbol{s}_g(k) z^k$ and $\boldsymbol{T}_g(z) = \sum_{k\geq 0} \boldsymbol{t}_g(k) z^k$ for the generating functions. We will also consider $\mathcal{C}_g(k,m)$ and $\mathcal{S}_g(k,m)$, which are respectively the collections of all chord diagrams and seeds with k chords and m 1-chords. Let $\boldsymbol{C}_g(x,y) = \sum_{k,m\geq 0} \boldsymbol{c}_g(k,m) x^k y^m$ and $\boldsymbol{S}_g(x,y) = \sum_{k,m\geq 0} \boldsymbol{s}_g(k,m) x^k y^m$ be the respective generating functions.

Theorem 3.2. (Andersen et al. [8]) *For any* $g \geq 1$ *we have the following relations.*

$$\begin{aligned}
\mathbf{C}_g(x,y) &= \frac{1}{x+y-xy}\mathbf{C}_g\left(\frac{x}{(1+x-xy)^2}\right),\\
\mathbf{S}_g(x,y) &= \frac{1+x}{1+2x-xy}\mathbf{C}_g\left(\frac{x(1+x)}{(1+2x-xy)^2}\right),\\
\mathbf{T}_g(z) &= z(1+2z)^{6g-2}P_g\left(\frac{z(1+z)}{(1+2z)^2}\right)\\
&= \sum_{j=2g}^{3g-1} p_g^{(j)} z^{j+1}(1+z)^j(1+2z)^{2(3g-1-j)},
\end{aligned}$$

where $P_g(z) = \sum_j p_g^{(j)} z^j$ *is as defined in theorem 3.1.*

We define an RNA σ-structure to be a partial chord diagram without any 1-chord, where all stacks have cardinality at least $\sigma \in \mathbb{N}$. The definition is chosen for its biological relevance; 1-chords are prohibited because of the tensile rigidity of the RNA sugar-phosphate backbone, and σ arises from the fact that stacks of small cardinalities are energetically unfavourable. Let $\mathcal{D}_\sigma(n)$ be the set of all RNA σ-structures on n vertices, and $\mathcal{D}_{g,\sigma}(n)$ be the subset consisting of structures with genus g, with the generating function

$$\mathbf{D}_{g,\sigma}(z) = \sum_{n\geq 0} \mathbf{d}_{g,\sigma}(n) z^n.$$

We have the following result for $\mathbf{D}_{g,\sigma}(z)$.

Theorem 3.3. (Andersen et al. [8]) *Suppose* $g, \sigma \geq 1$ *and let* $u_\sigma(z) = \frac{z^{2(\sigma-1)}}{z^{2\sigma}-z^2+1}$. *The generating function* $\mathbf{D}_{g,\sigma}$ *is given by*

$$\mathbf{D}_{g,\sigma}(z) = \frac{1}{u_\sigma(z)z^2 - z + 1} \mathbf{C}_g \left(\frac{u_\sigma(z) z^2}{(u_\sigma(z)z^2 - z + 1)^2} \right).$$

Let $\mathcal{P}(k)$ be the set of all partial chord diagrams with k chords. Any chord diagram can be obtained by removing isolated vertices from some partial chord diagram. In other words, we have a projection map

$$\varrho : \sqcup_{k\geq 0} \mathcal{P}(k) \to \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{C}_g(k),$$

which clearly preserves genus and the number of boundary components. Similarly, the reduction of each stack to a single chord defines a projection map

$$\vartheta : \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{C}_g(k) \to \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{S}_g(k),$$

which preserves genus, since at each stack, the reduction by a single chord results in the reduction of both the number of boundary components and the number of chords by 1. Finally, let $\mathcal{K}_g(k)$ be the set of all shadows of genus g on k chords. The removal of all non-crossing chords defines yet another genus-preserving projection

$$\varkappa : \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{S}_g(k) \to \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{K}_g(k).$$

Hence the composition

$$\pi = \varkappa \circ \vartheta \circ \varrho : \sqcup_{k\geq 0} \mathcal{P}(k) \to \sqcup_{g\geq 1} \sqcup_{k\geq 0} \mathcal{K}_g(k)$$

defines a genus-preserving projection from all RNA structures to the set of shadows. Furthermore, the genus of a shadow is equal to the sum of the genera of its irreducible components;

$$g(S) = g(\pi(S)) = \sum_{S' \text{ irred. component of } \pi(S)} g(S').$$

Therefore the natural filtering of RNA structure could be by the maximum genus in its irreducible components. For this we may introduce another class

of structures. An RNA γ-*structure* is a partial chord diagram S, such that the genera of the irreducible components of $\pi(S)$ is bounded by γ, i.e.

$$g(S') \leq \gamma \quad \forall S' \in \{S' \in \pi(S) \mid S' \text{ is an irreducible component in } \pi(S)\}.$$

Reidys et al. [77] have devised a folding algorithm for RNA γ-structures with $\gamma \leq 1$, based on the decomposition into irreducible shadows. This allowed an efficient implementation of topology-dependent energy penalties for pseudoknots. The resulting software achieved 10-20% increases in the prediction accuracy of base pairs.

Up to now the discussion has been on RNA structures with one backbone component. But the concepts of shadows, irreducibility and hence γ-structures apply equally well to the structures with more than one backbone components. In particular the analysis of structures over two backbone is relevant for the modelling of RNA-RNA interaction structures. A decomposition grammar for the γ-structures over two backbones with $\gamma = 0$ has been developed by Andersen et al. [7].

3.2 Recursion relation for RNA model

We will now discuss the enumeration of connected partial chord diagrams on multiple backbones, following the material presented in [3] and [15]. Let us consider a connected partial chord diagram, with b backbones, k chords, l marked points (unpaired vertices), and n boundary components. Note the diagram contains $2k + l$ vertices, of which $2k$ are 3-valent, and l are 2-valent. Let g be the genus of the diagram, which obeys Euler's formula,

$$b - k + n = 2 - 2g.$$

We then introduce the following combinatorial parameters.

- The *backbone spectrum* $\boldsymbol{b} = (b_0, b_1, \dots)$, where b_i is the number of backbones with i vertices (of degree either two or three).
- The *boundary point spectrum* $\boldsymbol{l} = (l_0, l_1, \dots)$, where l_i is the number of boundary components with i marked points. Note a marked point is contained in the boundary component that traverses above it.
- The *boundary length spectrum* $\boldsymbol{n} = (n_1, n_2, \dots)$, where n_K is the number of boundary components of length K. The length of a boundary component is defined to be the number of chords it traverses counted with multiplicity, plus the number of backbone underside it traverses. We note that a boundary component of length K can be divided into K *boundary segments*, by removing chords and backbone undersides. Each boundary segment contains zero or more marked points.
- The *boundary length and point spectrum* $\boldsymbol{m} = (m_{\boldsymbol{d}_K}, \dots)$, where $m_{\boldsymbol{d}_K}$ is the number of boundary components of length K with the *marked point spectrum* $\boldsymbol{d}_K = (d_1, \dots, d_K)$, which is defined to be the sequence of number of marked points in each boundary segment along a boundary component, modulo cyclic ordering. So in a boundary component with the marked point spectrum $\boldsymbol{d}_K = (d_1, \dots, d_K)$, we have d_1 marked points followed

by a chord or backbone underside, then d_2 marked points followed by a chord or backbone underside, and so on all the way around the boundary component.

Let $\boldsymbol{e}_j$ denote the sequence $(0,\dots,0,1,0,\dots)$, where 1 appears at the j'th entry and all other entries are 0. We say a diagram is of a certain type if it has the specified parameters and spectra; for example, a diagram of type $\{g,k,l;\boldsymbol{b},\boldsymbol{m}\}$ has genus g, k chords, l marked points, with the backbone spectrum $\boldsymbol{b}$ and the boundary length and point spectrum $\boldsymbol{m}$ (see figure 17 for an example).

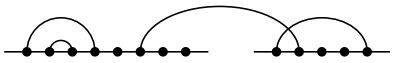


Figure 17: This chord diagram has 2 backbones, 4 chords, 5 marked points and 4 boundary components. By the Euler's formula, its genus is 0. It has the backbone spectrum $(\boldsymbol{e}_5+\boldsymbol{e}_8)$, the boundary point spectrum $(3\boldsymbol{e}_0+\boldsymbol{e}_5)$, the boundary length spectrum $(\boldsymbol{e}_1+2\boldsymbol{e}_2+\boldsymbol{e}_5)$, and the boundary length and point spectrum $(m_{(0)}=1, m_{(0,0)}=2, m_{(1,2,0,2,0)}=1)$.

The following relations follow immediately from the definitions.

$$\begin{aligned}
b &= \sum_{i\geq 0} b_i,\\
n = \sum_{i\geq 0} l_i &= \sum_{i\geq 1} n_i = \sum_{K\geq 1}\sum_{\boldsymbol{d}_K} m_{\boldsymbol{d}_K},\\
2k+l &= \sum_{i\geq 0} i b_i,\\
l = \sum_{i\geq 0} i l_i &= \sum_{K\geq 1}\sum_{\boldsymbol{d}_K} |\boldsymbol{d}_K|\, m_{\boldsymbol{d}_K},\\
2k+b = \sum_{i\geq 1} i n_i &= \sum_{K\geq 1}\sum_{\boldsymbol{d}_K} K m_{\boldsymbol{d}_K},\\
n_K &= \sum_{\boldsymbol{d}_K} m_{\boldsymbol{d}_K},\\
l_i &= \sum_{K\geq 1}\sum_{|\boldsymbol{d}_K|=i} m_{\boldsymbol{d}_K},
\end{aligned}$$

where $|\boldsymbol{d}_K| = \sum_{j=1}^{K} d_j$.

Let $\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l},\boldsymbol{n},\boldsymbol{m})$ be the number of distinct connected partial chord diagrams of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l},\boldsymbol{n},\boldsymbol{m}\}$. Note the parameters are not independent, as evident from the above relations. We set $\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l},\boldsymbol{n},\boldsymbol{m}) = 0$, if a partial chord diagram of the given type is not possible. We will also consider the number of distinct partial chord diagrams, where we sum one or more of the parameters, such as $\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l},\boldsymbol{n}) = \sum_{\boldsymbol{m}} \mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l},\boldsymbol{n},\boldsymbol{m})$ and $\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{m}) = \sum_{\boldsymbol{l}}\sum_{\boldsymbol{n}} \mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l},\boldsymbol{n},\boldsymbol{m})$. For a sequence indexed by integer such as $\boldsymbol{b}$, $\boldsymbol{l}$, and $\boldsymbol{n}$, and a variable $\boldsymbol{t} = (t_0, t_1, \dots)$, we denote

$$\boldsymbol{t}^{\boldsymbol{b}} = \prod_{i\geq 0} t_i^{b_i} = t_0^{b_0} t_1^{b_1} \cdots .$$

For the variable $\boldsymbol{m}$, indexed by ordered sets $\boldsymbol{d}_K = (d_1, \ldots, d_K)$, we consider the variable indexed by the same sets $\boldsymbol{u} = (u_{\boldsymbol{d}_K})$ and denote

$$\boldsymbol{u}^{\boldsymbol{m}} = \prod_{K \geq 1} \prod_{\boldsymbol{d}_K} u_{\boldsymbol{d}_K}^{m_{\boldsymbol{d}_K}}.$$

With these notations, the orientable, multi-backbone, boundary point spectrum generating function $F(x, y; \boldsymbol{s}; \boldsymbol{t})$ is given by

$$F(x, y; \boldsymbol{s}; \boldsymbol{t}) = \sum_{b \geq 1} F_b(x, y; \boldsymbol{s}; \boldsymbol{t}) \tag{31}$$

$$= \sum_{b \geq 1} \frac{1}{b!} \sum_{k=b-1}^{\infty} \sum_{\boldsymbol{l}} \sum_{\sum b_i = b} \mathcal{N}_{g,k,l}(\boldsymbol{b}, \boldsymbol{l}) x^{2g-2} y^k \boldsymbol{s}^{\boldsymbol{l}} \boldsymbol{t}^{\boldsymbol{b}}. \tag{32}$$

F satisfies the following differential equations. We also present its proof to illustrate the cut and join method.

Theorem 3.4. (Alexeev et al. [3]) *Consider the linear differential operators*

$$L_0 = \frac{1}{2} \sum_{i=0}^{\infty} \sum_{j=0}^{i} (i+2) s_j s_{i-j} \frac{\partial}{\partial s_{i+2}},$$

$$L_2 = \frac{1}{2} \sum_{i=2}^{\infty} s_{i-2} \sum_{j=1}^{i-1} j(i-j) \frac{\partial^2}{\partial s_j \partial s_{i-j}}$$

and the quadratic differential operators

$$QF = \frac{1}{2} \sum_{i=2}^{\infty} s_{i-2} \sum_{j=1}^{i-1} j(i-j) \frac{\partial F}{\partial s_j} \frac{\partial F}{\partial s_{i-j}}.$$

Then the following differential equations hold;

$$\begin{aligned} \tfrac{\partial F_1}{\partial y} &= (L_0 + x^2 L_2) F_1, \\ \tfrac{\partial F}{\partial y} &= (L_0 + x^2 L_2 + x^2 Q) F. \end{aligned} \tag{33}$$

These equations, together with the initial condition given by $x^{-2} \sum_{i \geq 1} s_i t_i$ for $y = 0$, determines the generating functions F_1 and F uniquely.

Proof. We note first that equation (33) is equivalent to the following recursion

relation for $\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l})$:

$$
\begin{aligned}
&k\mathcal{N}_{g,k,l}(\boldsymbol{b},\boldsymbol{l}) = \\
&\quad \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=0}^{i}(i+2)(l_{i+2}+1)\mathcal{N}_{g,k-1,l+2}(\boldsymbol{b},\boldsymbol{l}-\boldsymbol{e}_j-\boldsymbol{e}_{i-j}+\boldsymbol{e}_{i+2}) + \\
&\quad \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=1}^{i+1} j(i+2-j)(l_j+1+\delta_{j,i+2-j}-\delta_{i,j})(l_{i+2-j}+1-\delta_{j,2}) \times \\
&\qquad \mathcal{N}_{g-1,k-1,l+2}(\boldsymbol{b},\boldsymbol{l}+\boldsymbol{e}_j+\boldsymbol{e}_{i+2-j}-\boldsymbol{e}_i) + \\
&\quad \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=1}^{i+1}\sum_{g_1+g_2=g}\sum_{k_1+k_2=k-1}\sum_{\boldsymbol{l}^{(1)}+\boldsymbol{l}^{(2)}=\boldsymbol{l}-\boldsymbol{e}_i}\sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}} \\
&\qquad j(i+2-j)(l_j^{(1)}+1)(l_{i+2-j}^{(2)}+1)\frac{b!}{b^{(1)}!b^{(2)}!} \times \\
&\qquad \mathcal{N}_{g_1,k_1,l_1+j}(\boldsymbol{b}^{(1)},\boldsymbol{l}^{(1)}+\boldsymbol{e}_j)\mathcal{N}_{g_2,k_2,l_2+i+2-j}(\boldsymbol{b}^{(2)},\boldsymbol{l}^{(2)}+\boldsymbol{e}_{i+2-j}), \qquad (34)
\end{aligned}
$$

where

$$b^{(r)} = \sum_{i=1}^{\infty} b_i^{(r)}, \quad l_r = \sum_{i=1}^{\infty} i l_i^{(r)}, \quad \sum_{i=0}^{\infty} l_i^{(r)} = k_r - 2g_r - b^{(r)} + 2,$$

for $r = 1, 2$. Checking this is a straightforward computation looking at the coefficients of $x^{2-2g}y^{k-1}\boldsymbol{s}^{\boldsymbol{l}}\boldsymbol{t}^{\boldsymbol{b}}$ in both sides of equation (33). The recursion relation (34) is then proved by the cut and join method, similarly to theorem 4.1. The idea is to express the number of diagrams of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l}\}$ with one marked chord in two different ways, one by simply marking a chord on a diagram of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l}\}$, and the other by adding a marked chord to a diagram, such that the resulting diagram is of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l}\}$. The former is straightforward; given a diagram of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l}\}$, there are k chords to choose from for marking, so this gives the l.h.s. in equation (34). We do the latter by first removing a chord from a diagram of type $\{g,k,l,\boldsymbol{b},\boldsymbol{l}\}$ ("cut"), then adding an appropriate marked chord ("join"). See theorem 4.1 for more details in proof. □

The recursion relations for other spectra, such as the boundary length spectrum (for chord diagrams) [3], and the boundary length and point spectrum (for partial chord diagrams) [15] have been established using the same method.

Recursion relation for non-oriented chord diagrams

Non-oriented (partial) chord diagrams can be considered by assigning an extra binary data to each chord, showing whether the chord is twisted or not. If C is such a (partial) chord diagram, on the associated surface $F = F(C)$, the twisted chord then become twisted bands. Clearly this construction produces 2^k different orientable and non-orientable partial chord diagrams from one orientable partial chord diagram with k chords. We have the following definition for the Euler characteristic in the non-oriented case. The *Euler characteristic* χ of the non-oriented surface F is give by

$$\chi(F) = 2 - h,$$

where h is the number of cross-caps and we have Euler's relation

$$2 - h = b - k + n.$$

We also need to introduce a small change for the boundary point and length spectrum $\boldsymbol{m}$. Since we are no longer able to traverse a boundary component using the induced orientation in the non-orientable case, we also need to consider the order-reversing, as well as cyclic, permutation. So we have

$$\boldsymbol{d}_K = (d_1, d_2, \ldots, d_K) = (d_K, \ldots, d_2, d_1).$$

With these definitions, it is possible to derive the recursion relations for the number of orientable and non-orientable (partial) chord diagrams, filtered by different spectra. The specific expressions and derivations can be found in [3, 15].

3.3 Enumeration of RNA chord diagrams via matrix models

Let us recall and modify the (formal) matrix integral defined in section 2.3 to enumerate RNA linear chord diagrams, as discussed in [12]. The potential for the model is

$$V(M) = \frac{M^2}{2} - \frac{stM}{1 - tM} = \frac{M^2}{2} - s\sum_{k\geq 1}(tM)^k, \tag{35}$$

with the partition function

$$\log Z(s, t, N) = \frac{1}{Z_0}\int \mathrm{d}M e^{-N\mathrm{Tr}V(M)} = \sum_{\tau} \frac{N^{\chi(\tau)} s^{b(\tau)} t^{2n(\tau)}}{\#\mathrm{Aut}(\tau)}, \tag{36}$$

where $\chi(\tau), b(\tau)$ and $n(\tau)$ denote respectively the Euler characteristic, the number of backbones, and the number of chords for a chord diagram τ, and the sum is over all connected chord diagrams. Indeed, compared to equation (9), the variables t_j there are replaced by a single variable s, thus tracking the total number of backbones (instead of the number for each valency), and the variable t here tracks the total number of half-edges, which is equal to twice the number of chords in a linear chord diagram.

In [12], Andersen, Chekhov, Penner, Reidys and Sułkowski applied the topological recursion framework (section 2.4) to solve the matrix model for RNA linear chord diagrams. The challenge, in some ways, lies in the determination of the spectral curve. We will not reproduce the details here, but in this particular case the cut end points were determined as perturbative expansions in variables s and t, and some scaling of variables were required to obtain more manageable expression for the spectral curve. In [13], the results were generalised for the RNA matrix model for non-oriented linear chord diagrams using so-called the β-deformed topological recursion [27].

To construct a matrix model for partial chord diagrams, we need external matrices Λ_{P} to represent unpaired half-edges, which do not participate in Wick contractions. We present the construction below, following the materials presented in [14].

Let us first consider the number of partial chord diagrams filtered by the boundary point spectrum $\boldsymbol{l} = (l_0, l_1, \dots)$, where l_i is the number of boundary components with i marked points (a marked point is contained in the boundary component that traverses above it). What we require is a bijective correspondence between partial chord diagrams and sets of Wick contractions. This is given by the following set of rules, which is illustrated in figure 18. Let C be a partial chord diagram and $F(C)$ the associated surface. Let Λ_{P} be an $N \times N$ matrix. We assign matrix elements to $F(C)$ by the following rules [14];

P1: A matrix element $M_{\alpha\beta}$ is assigned to a chord end on a backbone. Indices $\alpha, \beta = 1, 2, \dots, N$ are assigned to the two vertical segment by each vertex along a backbone.

P2: A matrix element $\Lambda_{\mathrm{P}\alpha\beta}$ is assigned to a marked point.

P3: Suppose, after applying rules **P1** and **P2**, $U_{\alpha_j\beta_j}$ and $V_{\alpha_{j+1}\beta_{j+1}}$ are adjacent chord ends or marked points on the same backbone with $U, V = M$ or Λ_{P}. Then we assign the sum

$$\sum_{\beta_j, \alpha_{j+1}=1}^{N} \delta_{\beta_j\alpha_{j+1}}$$

to the horizontal boundary segment between them.

P4: Suppose, after applying **P3**, we have a matrix product

$$(M^{v_1}\Lambda_{\mathrm{P}}^{w_1} M^{v_2}\Lambda_{\mathrm{P}}^{w_2} \cdots M^{v_i}\Lambda_{\mathrm{P}}^{w_i})_{\alpha_1\beta_i}, \quad v_j, w_j \in \mathbb{Z}_{\geq 0}, \ \sum_{j=1}^{i}(v_j + w_j) = i$$

corresponding to a backbone with i vertices. Then we assign

$$N \sum_{\alpha_1, \beta_i=1}^{N} \delta_{\beta_i,\alpha_1}$$

to the bottom edge of this backbone. It follows, by going around the backbone, that the trace

$$N\mathrm{Tr}(M^{v_1}\Lambda_{\mathrm{P}}^{w_1} M^{v_2}\Lambda_{\mathrm{P}}^{w_2} \cdots M^{v_i}\Lambda_{\mathrm{P}}^{w_i})$$

is assigned to this backbone.

P5: To a band connecting $M_{\alpha_j\beta_j}$ and $M_{\alpha'_{j'},\beta'_{j'}}$, we assign a Wick contraction

$$N \left\langle M_{\alpha_j\beta_j} M_{\alpha'_{j'},\beta'_{j'}} \right\rangle = \delta_{\alpha_j,\beta'_{j'}}\delta_{\beta_j,\alpha'_{j'}}.$$

Note, all possible traces $N\mathrm{Tr}(M^{v_1}\Lambda_{\mathrm{P}}^{w_1} \cdots M^{v_i}\Lambda_{\mathrm{P}}^{w_i})$ on a backbone with i vertices are generated by $N\mathrm{Tr}(M + \Lambda_{\mathrm{P}})^i$. Hence, given a backbone spectrum $\boldsymbol{b}$, all possible sequences of M and Λ_{P} on $\boldsymbol{b}$ are generated by the product

$$\prod_{i\geq 0} \left(N\mathrm{Tr}(M + \Lambda_{\mathrm{P}})^i\right)^{b_i}.$$

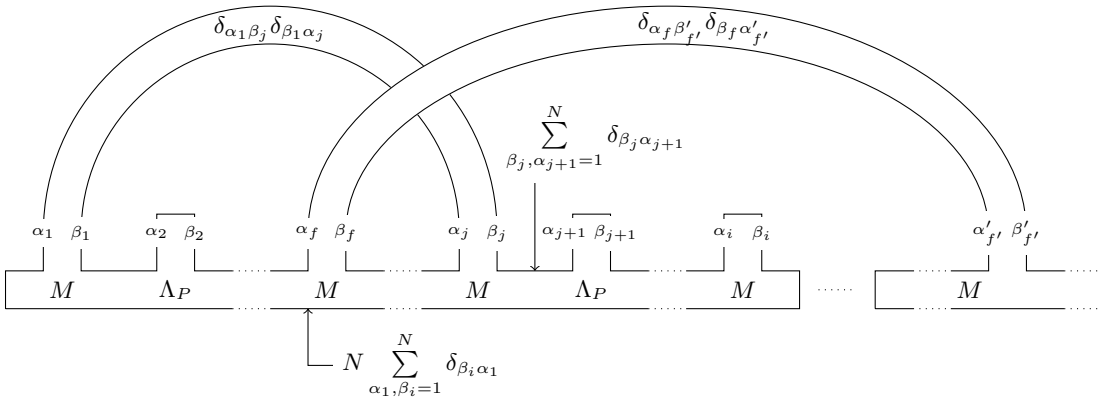


Figure 18: Assignment of matrix elements to a partial chord diagram.

The above rules imply that all partial chord diagrams with the backbone spectrum $\boldsymbol{b}$ correspond bijectively to all matchings among M's in the expansion of the Gaussian average

$$W_N^P(\boldsymbol{b},\boldsymbol{r}) = \left\langle \prod_{i\geq 0}\left(N\mathrm{Tr}(M+\Lambda_{\mathrm{P}})^i\right)^{b_i}\right\rangle, \tag{37}$$

where $\boldsymbol{r}=(r_1,r_2,\cdots)$ with $r_i=\frac{1}{N}\mathrm{Tr}\Lambda_{\mathrm{P}}^i$. In a chord diagram with the boundary point spectrum $\boldsymbol{l}$, there are l_i boundary components with i marked points. Each of the l_i boundary components contributes to $W_N^P(\boldsymbol{b},\boldsymbol{r})$ with the factor

$$\sum_{\alpha_{j_1},\ldots,\alpha_{j_i}=1}^{N}\Lambda_{\mathrm{P}\alpha_{j_1}\alpha_{j_2}}\Lambda_{\mathrm{P}\alpha_{j_2}\alpha_{j_3}}\cdots\Lambda_{\mathrm{P}\alpha_{j_i}\alpha_{j_1}}=\mathrm{Tr}\Lambda_{\mathrm{P}}^i,$$

so the total contribution is with the factor $(\mathrm{Tr}\Lambda_{\mathrm{P}}^i)^{l_i}$. It follows, for partial chord diagrams with the backbone spectrum $\boldsymbol{b}$ and the boundary point spectrum $\boldsymbol{l}$, the corresponding term in $W_N^P(\boldsymbol{b},\boldsymbol{r})$ has the factor

$$N^{b-k+l_0}\prod_{i\geq 0}\left(\mathrm{Tr}\Lambda_{\mathrm{P}}^i\right)^{l_i}=N^{b-k+n}\prod_{i\geq 0}r_i^{l_i}.$$

Note we get l_0 in the power of N, since a Wick contraction $\left\langle M_{\alpha_j\beta_j}M_{\alpha'_{j'}\beta'_{j'}}\right\rangle$ contributes with a factor N if and only if the boundary component formed by the corresponding band does not contain any marked point. Thus we obtain,

Proposition 3.5. (Andersen et al. [14]) *The Gaussian average (37) is the generating function for the numbers $\widehat{\mathcal{N}}_{k,b,l}(\boldsymbol{b},\boldsymbol{r})$ of connected and disconnected partial chord diagrams with the backbone spectrum $\boldsymbol{b}$ and the boundary point spectrum $\boldsymbol{l}$;*

$$W_N^P(\boldsymbol{b},\boldsymbol{r})=\sum_{\boldsymbol{l}}\widehat{\mathcal{N}}_{k,b,l}(\boldsymbol{b},\boldsymbol{r})N^{b-k+n}\prod_{i\geq 0}r_i^{n_i},$$

where the summation is constrained by $\sum in_i=\sum ib_i-2k$.

With the above proposition, we wish to express the orientable, multi-backbone, boundary point spectrum generating function using matrix integral. Recall the generating function for connected diagrams given in equation (32). The corresponding generating function for connected and disconnected diagrams is given by the exponential;

$$\begin{aligned} Z^{\mathrm{P}}(x,y;\boldsymbol{s};\boldsymbol{t}) &= \exp\left(F(x,y;\boldsymbol{s};\boldsymbol{t})\right) \\ &= \sum_{\boldsymbol{b}}\sum_{\boldsymbol{l}} \widehat{\mathcal{N}}_{k,b,l}(\boldsymbol{b},\boldsymbol{l}) x^{-b+k-n} y^k \prod_{i\geq 0} \boldsymbol{s}^{\boldsymbol{l}}\boldsymbol{t}^{\boldsymbol{b}}. \end{aligned} \tag{38}$$

On the other hand, the full generating function with variables $\boldsymbol{s}$ and $\boldsymbol{r}$ is given by

$$\begin{aligned} Z_N^{\mathrm{P}}(y;\boldsymbol{t};\boldsymbol{r}) &= \sum_{\boldsymbol{b}}\sum_{\boldsymbol{l}} \widehat{\mathcal{N}}_{k,b,l}(\boldsymbol{b},\boldsymbol{l}) N^{b-k+n} y^k \prod_{i\geq 0} \boldsymbol{t}^{\boldsymbol{b}}\boldsymbol{r}^{\boldsymbol{l}} \\ &= \sum_{\boldsymbol{b}} y^{\sum_i b_i/2} W_N^{\mathrm{P}}(\boldsymbol{b}; y^{-i/2}\boldsymbol{r}) \prod_i \frac{s_i^{b_i}}{b_i!} \\ &= \sum_{\boldsymbol{b}} \prod_{i\geq 0} \frac{s_i^{b_i} y^{ib_i/2}}{b_i!} \left\langle \left(N\mathrm{Tr}(M+y^{-1/2}\Lambda_{\mathrm{P}})^i\right)^{b_i} \right\rangle, \end{aligned}$$

where $b_i!$ in the denominator comes from the fact that the number of partial chord diagrams is invariant under permutations of its backbones. Doing the summation over $\boldsymbol{b}$, we find that Z_N^{P} is given by the (formal) matrix integral

$$\begin{aligned} Z_N^{\mathrm{P}}(y;\boldsymbol{t};\boldsymbol{r}) &= \prod_{i\geq 0} \exp\left(s_i y^{i/2} \left\langle N\mathrm{Tr}(M+y^{-1/2}\Lambda_{\mathrm{P}})^i \right\rangle \right) \\ &= \frac{1}{\mathrm{Vol_N}} \int_{\mathcal{H}_N} \mathrm{d}M \exp\left(-N\mathrm{Tr}\left(\frac{M^2}{2} - \sum_{i\geq 0} s_i (y^{1/2}M+\Lambda_{\mathrm{P}})^i \right) \right). \end{aligned} \tag{39}$$

Finally, we identify $Z^{\mathrm{P}}(x,y;\boldsymbol{s};\boldsymbol{t})$ and $Z^{\mathrm{P}_N}(y;\boldsymbol{t};\boldsymbol{r})$ by a change of variables. For $i=0$, we have

$$r_0 = \frac{1}{N}\mathrm{Tr}\Lambda_{\mathrm{P}}^0 = 1,$$

so we are forced to take the following change of variables;

$$N \to t_0 N, \quad y \to t_0 y, \quad , \boldsymbol{s} \to t_0^{-1}\boldsymbol{s}, \quad \boldsymbol{r} \to t_0^{-1}\boldsymbol{t}.$$

We have now proved,

Theorem 3.6. (Andersen et al. [14]) *The generating function $Z^{\mathrm{P}}(x,y;\boldsymbol{s};\boldsymbol{t})$ and the matrix integral $Z_N^{\mathrm{P}}(y;\boldsymbol{t};\boldsymbol{r})$ satisfy*

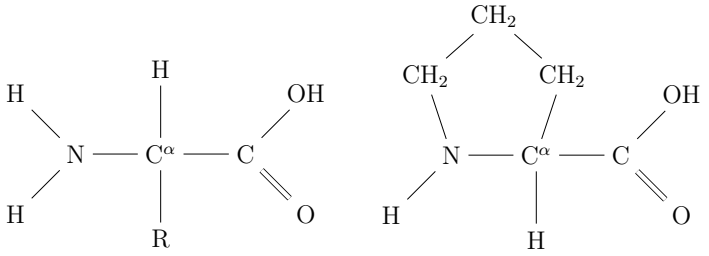
$$Z^{\mathrm{P}}(x,y;\boldsymbol{s};\boldsymbol{t}) = Z_{t_0 N}^{\mathrm{P}}(t_0 y; t_0^{-1}\boldsymbol{s}; t_0^{-1}\boldsymbol{t}).$$

Relations corresponding to the other spectra listed in section 3.2 can also be found in [14].

4 Protein Model

4.1 Structure of proteins

A protein is a linear polymer of amino acids, of which there are 20 different kinds. All but one of the 20 amino acids have a standard structure, as shown in figure 19a. It is the so-called residues (marked with "R") connected to the α-carbon atom (marked with "C^{α}"), which characterise amino acids. The last amino acid proline has a slightly different structure, containing a ring CCCCN (figure 19b). Proline occurs relatively infrequently in proteins, although they have a structural importance for their association with turns in protein structure. We will nonetheless use the typical amino acid structure shown in figure 19a to depict all amino acids in the following text, keeping in mind that some of them may be replaced by a proline. The OH in the right-hand side



(a) Typical amino acid (b) Proline

Figure 19: Chemical structure of amino acids

and H in the left-hand side of another amino acid can combine to produce a water molecule, resulting in a covalent bond called a peptide bond between C and N atoms. A chain of multiple amino acids combined in this manner is called a polypeptide (figure 20). A polypeptide is oriented from the N- to the C-terminus by convention. Now let $\mathrm{N_i}$, $\mathrm{C_i^{\alpha}}$, $\mathrm{C_i}$, $\mathrm{R_i}$, $\mathrm{O_i}$ denote respectively the $\mathrm{N}, \mathrm{C}^{\alpha}, \mathrm{C}, \mathrm{R}, \mathrm{O}$ in the i'th amino acid in a polypeptide, $i \in \{1, 2, \ldots, L\}$, where L is the length of a polypeptide measured in the number of amino acids. Let also $\mathrm{H_i}$ denote the H atom connected to $\mathrm{N_i}$, $i \in \{2, 3, \ldots, L\}$. Then the i'th *peptide unit* of a polypeptide consists of the six atoms around the i'th peptide bond $(\mathrm{C_i^{\alpha}}, \mathrm{O_i}, \mathrm{C_i}, \mathrm{N_{i+1}}, \mathrm{H_{i+1}}, \mathrm{C_{i+1}^{\alpha}})$, and we say two consecutive peptide units are connected by an *α-carbon links*. We will make the following assumptions, as was done in [72] and are generally accepted to hold for the polypeptide structures.

Assumptions. *Let* $\mathrm{N_i}$, $\mathrm{C_i^{\alpha}}$, $\mathrm{C_i}$, $\mathrm{R_i}$, $\mathrm{O_i}, \mathrm{H_i}$ *as above. We have;*

- *Six atoms in a peptide unit* $(\mathrm{C_i^{\alpha}}, \mathrm{O_i}, \mathrm{C_i}, \mathrm{N_{i+1}}, \mathrm{H_{i+1}}, \mathrm{C_{i+1}^{\alpha}})$ *lie on a common plane.*
- *The angles around* $\mathrm{C_i}$ *and* $\mathrm{N_{i+1}}$ *that form a peptide bond are* $120°$.
- *The angles around the α-carbon atoms are tetrahedral.*
- *In each peptide unit, the centres of the two alpha-carbon atoms lie on either side of the line determined by the peptide bond, except occasionally for the peptide unit preceding proline.*

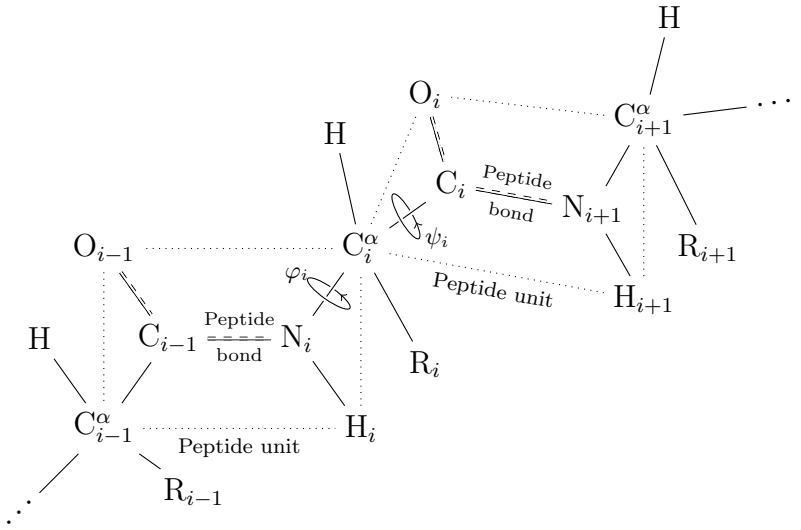
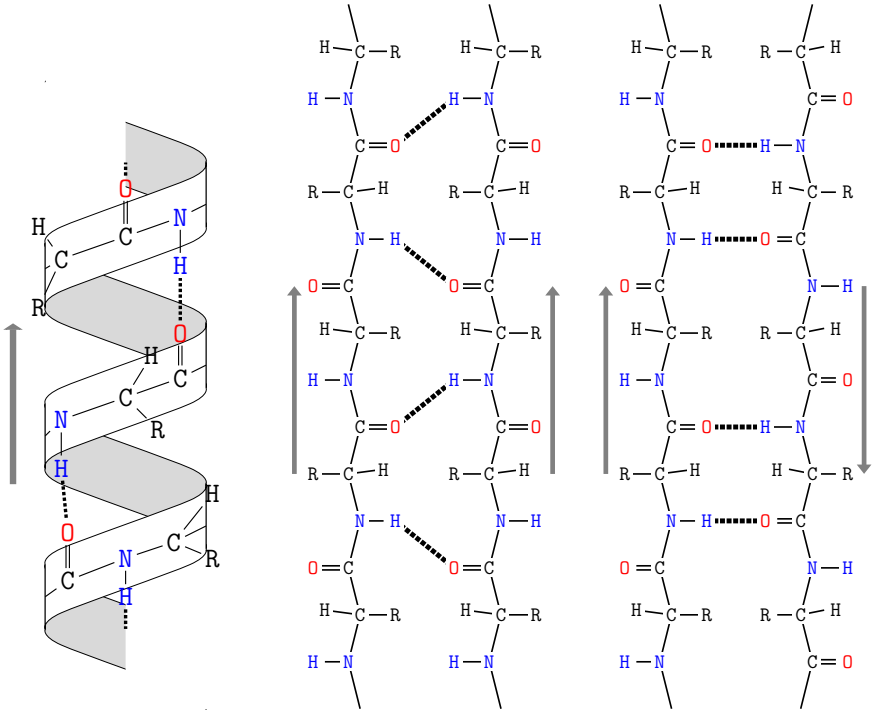


Figure 20: Chemical structure of polypeptide

Note these geometric constraints are not mathematical in nature; they must be understood as having some flexibility; i.e. "nearly" lying on a common plane, etc. The last point expresses the dominance of so-called *trans-conformation*. The complementary possibility is called the *cis-conformation*, which, as stated, occurs infrequently. The pairs of rotation angles φ and ψ around the $\mathrm{N_i} - \mathrm{C_i^{\alpha}}$ and $\mathrm{C_i^{\alpha}} - \mathrm{C_i}$ axes are called *conformational angles*, and the sequence of atoms

$$\mathrm{N}_1 - \mathrm{C}_1^{\alpha} - \mathrm{C}_1 - \mathrm{N}_2 - \mathrm{C}_2^{\alpha} - \cdots - \mathrm{N}_\mathrm{L} - \mathrm{C}_\mathrm{L}^{\alpha} - \mathrm{C}_\mathrm{L}$$

in a polypeptide with L amino acids is called its *backbone*. The *primary structure* of a polypeptide is the sequence of its constituent residues ordered from the N- to C-terminus, which can also be thought of as a word in an alphabet with 20 letters. H_i in one peptide unit and O_j in another may form a hydrogen bond, in which case the N_i atom is called the *donor* and O_j the *acceptor* of the hydrogen bond. By the *H-graph structure* of a protein we mean its primary structure together with all its hydrogen bonds. The *secondary structure* of a protein is the collection of its local structures, categorised into common patterns such as α-helix or β-sheet. An α-helix is a structure where the backbone forms a right-handed helix with every amino hydrogen bonded to a carboxyl oxygen with three residues in between (figure 21a). A β-sheet consists of two or more β-strands connected laterally by hydrogen bonds, forming a sheet. Two adjacent β-strands in a β-sheet may be arranged in a parallel (figure 21b) or antiparallel configuration (figure 21c), depending on the orientations of the two backbones. The folded, 3-dimensional structure of a protein is called its *tertiary structure*, and it is determined by the primary and secondary structures, along with various other weaker forces and interactions, such as the van der Waals forces and the hydrophobicity of the residues, to name two.



(a) α-helix (b) Parallel (c) Anti-parallel

Figure 21: Schematic representations of α-helix (21a), parallel β-sheet (21b) and anti-parallel β-sheet (21c). Hydrogen bonds are shown in dotted lines. The arrows indicate backbone orientations.

4.2 The protein model

Let us consider the standard structure of a peptide unit. Our model is the one described in [72], which is a very natural abstraction of the standard structure, where the backbone is shown as a horizontal line from C_i^α to C_{i+1}^α, with the O_i and H_{i+1} atoms shown as the edge above and below the backbone edge, respectively (figure 22a). The edges below the backbone are denoted by N instead of H, to represent the donor-acceptor relation described above. These building blocks are concatenated to build a model of the given backbone (figure 23). Note the positions of O- and N-edges correspond to those of trans isomers. In a peptide unit preceding cis-proline, the more accurate representation would be to have O- and N-edges on the same side of the backbone (figure 22b). Even though our model allows for such a representation of cis-proline, they are known to relatively rare [2]. We will therefore use a standard, single building block here for our model for the ease of computation. The inclusion of a special cis-proline building block is certainly a possibility for a future investigation. A hydrogen bond is represented by an edge (which we call *chords*, in keeping with the other literature on RNA modelling, e.g. [3, 15]) between the corresponding donor (N-) and acceptor (O-) half-edges (figure 24). This means that the chords always have one endpoint in the upper half-plane and the other endpoint in the lower

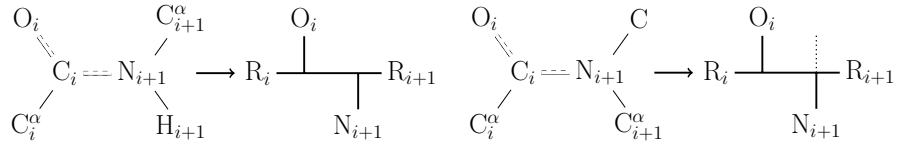


Figure 22: Model of a standard peptide unit (a) and of a peptide unit preceding cis-proline (b). The more geometrically accurate position of N half-edge in is shown by the dotted line in (b).

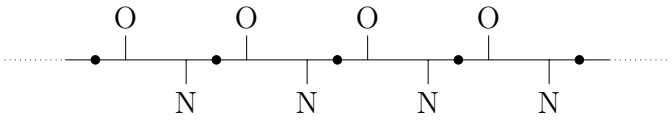


Figure 23: Model of a backbone.

half-plane. Note the orientation of the backbone from the N- to O- terminus and of the hydrogen bond from the N-donor to the O-acceptor.

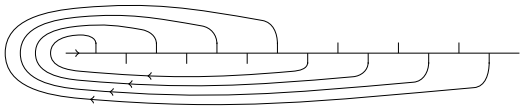


Figure 24: Model of α-helix structure.

A natural generalisation of this model is to include the information on rotational angles at α-carbon links or hydrogen bonds. We do this by following the method in [72], which we briefly describe here. First we associate a three-frame $\mathcal{F}_i = (u, v, w),\ u, v, w \in \mathbb{R}^3$ to the i'th peptide unit P_i, $i = 1, 2, \ldots$ as follows. u is the unit vector parallel to the displacement vector $\overrightarrow{\mathrm{C}_i\mathrm{N}_{i+1}}$ along the peptide bond. Let $u^\perp$ be the vector in the plane of peptide unit which is perpendicular to u. Let $\bar{v}$ be the projection of the vector $\overrightarrow{\mathrm{C}^\alpha_i\mathrm{C}_i}$ onto the subspace $\mathbb{R}u^\perp$ and set $v = \bar{v}/\|\bar{v}\|$. Finally, we set $w = u \times v$. Let $\mathcal{F}_j$ be the three frame associated to another peptide unit P_j. Then there is a unique element of the group SO(3) which takes $\mathcal{F}_i$ to $\mathcal{F}_j$. However, this element of SO(3) is not invariant under rotation of the entire protein. The solution to this problem is to rotate the entire protein so that $\mathcal{F}_i$ becomes the standard three-frame in $\mathbb{R}^3$. If $A_i \in \mathrm{SO}(3)$ is the matrix with columns consisting of u, v, and w, and A_j corresponds to $\mathcal{F}_j$ in the same way, then $R_{ij} = A_i^{-1}A_j \in \mathrm{SO}(3)$ maps the identity I to $A_i^{-1}A_j$. If there is an edge (α-carbon link or hydrogen bond) from P_i to P_j in our model, we assign $R_{ij} \in \mathrm{SO}(3)$ to this edge. The rotation information at each α-carbon link and hydrogen bond is then discretised to a binary decoration consisting of values "twisted" or "untwisted" by seeing which of the associated $R_{ij} \in \mathrm{SO}(3)$ or its "flipped" counterpart $\hat{R}_{ij}$ lies closest to the identity, with respect to the unique bi-invariant metric on SO(3). More precisely, write $R_{ij} = (u, v, w)$, where u, v, w are vectors in $\mathbb{R}^3$. Set $\hat{R}_{ij} = (u, -v, -w)$. This is the element of SO(3) that corresponds to taking $\mathcal{F}_i$ to $\hat{\mathcal{F}}_j$, which is $\mathcal{F}_j$ turned upside down by

rotating it by 180° about the line containing C_j and N_{j+1}. The edge (α-carbon link or hydrogen bond) between the i'th and j'th peptide units is twisted, if and only if $d(I, R_{ij}) \geq d(I, \hat{R}_{ij})$, where d is the unique bi-invariant metric on SO(3). Clearly the resulting fatgraph model may not be orientable, depending on the number of twisted edges. It is nonetheless possible to define topological invariants such as the number of boundary components and genus (section 2.1). These invariants have been successfully used to classify local structures within proteins [74].

In this thesis, unless otherwise specified, we will consider the undecorated, orientable fatgraph model of proteins. The choice was made so that we can utilise the computational efficiency of the expression of fatgraphs as a pair of permutations (section 2.1), and as a purely combinatorial and discrete object, it was expected be simpler to work with.

4.3 Recursion relation for the protein model

Since our model of proteins, like the RNA model, is based on fatgraphs, it is natural to expect that it satisfies recursion relations similar to the ones described in section 3.2. An important difference is that we now have two types of half-edges; O and N half-edges, with an extra condition that O-O and N-N propagators are not allowed. We therefore require a way of tracking presence of O and N half-edges around the backbone. Let us describe how it can be done, and present the recursion relation.

As in section 3.2, we let g, k, l, n denote respectively the genus, the number of chords, the number of marked points (i.e. unbonded O and N half edges), and the number of boundary components, of a protein diagram. We also define one further combinatorial parameter.

- The *boundary type spectrum* $\boldsymbol{q} = (q_{\boldsymbol{p}_K}, \dots)$, where $q_{\boldsymbol{p}_K}$ is the number of boundary components with the *type signature* $\boldsymbol{p}_K = (p_1, p_2, \dots, p_K)$, which is a sequence of $p_i \in \{O, N\}$, showing the type of marked points along the boundary component.

As an example, consider figure 24. We have $l = 6$, $k = 4$, $n = 5$, so $g = 0$. The backbone spectrum is $\boldsymbol{e}_{14}$, and the boundary type spectrum is $(q_{()} = 3, q_{(NNN)} = 1, q_{(OOO)} = 1)$.

We will consider the single-backbone case, so we have $\boldsymbol{b} = \boldsymbol{e}_{2k+l}$. Let $\mathcal{N}_{g,k,l}(\boldsymbol{e}_{2k+l}; \boldsymbol{q})$ be the number of distinct diagrams of type $\{g, k, l, \boldsymbol{e}_{2k+l}, \boldsymbol{q}\}$.

Theorem 4.1. *$\mathcal{N}_{g,k,l}(\boldsymbol{e}_{2k+l}; \boldsymbol{q})$ satisfies the following recursion relation;*

$$
\begin{aligned}
k\mathcal{N}_{g,k,l}(\boldsymbol{e}_{2k+l}; \boldsymbol{q}) = &\sum_{K\geq 1} \sum_{\boldsymbol{p}_K} (q_{\boldsymbol{p}_K} + 1) \\
&\sum_{\substack{I<J\leq K \\ p_I \neq p_J}} \sum_{1\leq I\leq K} \mathcal{N}_{g,k-1,l+2}(\boldsymbol{e}_{2k+l}; \boldsymbol{q} + s_{I,J}(\boldsymbol{p}_K)) \\
&+ \frac{1}{2} \sum_{K=1}^{\infty} \sum_{L=1}^{\infty} \sum_{\boldsymbol{p}_K} \sum_{\boldsymbol{r}_L} (q_{\boldsymbol{p}_K} + 1)(q_{\boldsymbol{r}_L} + 1 + \delta_{\boldsymbol{p}_K, \boldsymbol{r}_L}) \\
&\sum_{\substack{1\leq I\leq K \\ p_I \neq r_J}} \sum_{1\leq J\leq L} \mathcal{N}_{g-1,k-1,l+2}(\boldsymbol{e_{2k+l}}; \boldsymbol{q} + t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L)),
\end{aligned}
$$

where

$$s_{I,J}(\boldsymbol{p}_K) = \boldsymbol{e}_{\boldsymbol{p}_K} - \boldsymbol{e}_{(p_1 \dots p_{I-1} p_{J+1} \dots p_K)} - \boldsymbol{e}_{(p_{I+1} p_{I+2} \dots p_{J-2} p_{J-1})}$$

and

$$t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L) = \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} - \boldsymbol{e}_{(p_1 \dots p_{I-1} r_{J+1} \dots r_L r_1 \dots r_{J-1} p_{I+1} \dots p_K)}$$

Proof. As described in theorem 3.4, we will express the number of diagrams of type $\{g, k, l, \boldsymbol{e}_{2k+l}, \boldsymbol{q}\}$ in two different ways; one by removing and adding a marked chord, and the other by simply marking a chord. For removing a chord, there are two cases to consider; the first is when the chord to be removed is part of two distinct boundary components, and the second is when it is part of a single boundary component.

For the first case, when we remove the chord the two boundary components which the removed chord was part of are merged to one. The resulting diagram therefore has $k-1$ chords and $n-1$ boundary components, hence the genus g is unchanged. It has $l+2$ marked points, since the two end points of the removed chord become marked points. Suppose the merged boundary component has the signature $\boldsymbol{p}_K = (p_1 p_2 \dots p_K)$. Then prior to merging the signatures $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ of two boundary components are

$$\begin{aligned} \boldsymbol{p}_1 &= (p_1 p_2 \dots p_{I-1} p_{J+1} p_{J+2} \dots p_K), \\ \boldsymbol{p}_2 &= (p_{I+1} p_{I+2} \dots p_{J-2} p_{J-1}), \end{aligned}$$

for some $1 \le I < J \le K$ with the condition $p_I \neq p_J$. So if the diagram had the boundary point and type spectrum $\boldsymbol{q}$ prior to the chord removal, then its spectrum after the removal is given by

$$\boldsymbol{q} - \boldsymbol{e}_{\boldsymbol{p}_1} - \boldsymbol{e}_{\boldsymbol{p}_2} + \boldsymbol{e}_{\boldsymbol{p}_K} = \boldsymbol{q} + s_{I,J}(\boldsymbol{p}_K).$$

In order to obtain a diagram of type $\{g, k, l; \boldsymbol{e}_{2k+l}; \boldsymbol{q}\}$ with a marked chord, we can add a marked chord to one of the $q_{\boldsymbol{p}_K} + 1$ boundary components formed as the result of a chord removal, so the total number of diagrams in this case is

$$\sum_{K \ge 1} \sum_{\boldsymbol{p}_K} (q_{\boldsymbol{p}_K} + 1) \sum_{\substack{1 \le I \le K \\ p_I \neq p_J}} \sum_{I < J \le K} \mathcal{N}_{g,k-1,l+2}(\boldsymbol{e}_{2k+l}; \boldsymbol{q} + s_{I,J}(\boldsymbol{p}_K)). \tag{40}$$

In the second case, removing the chord splits the boundary component in two. So after the removal there are $k-1$ chords and $n+1$ boundary components, hence the genus becomes $g-1$. If we suppose the signatures of the two boundary components after the removal are $\boldsymbol{p}_K$ and $\boldsymbol{r}_L$, the original boundary component had the signature

$$\tilde{\boldsymbol{p}} = (p_1 \dots p_{I-1} r_{J+1} \dots r_L r_1 \dots r_{J-1} p_{I+1} \dots p_K),$$

for $1 \le I \le K$ and $1 \le J \le L$ with the condition $p_I \neq r_J$. If the spectrum prior to the removal was $\boldsymbol{q}$, then after the removal it becomes

$$\boldsymbol{q} - \boldsymbol{e}_{\tilde{\boldsymbol{p}}} + \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} = \boldsymbol{q} + t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L).$$

Now there are $(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1)$ choice of boundary components, if $\boldsymbol{p}_K \neq \boldsymbol{r}_L$, and $(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{p}_K}+2)/2$ if $\boldsymbol{p}_K = \boldsymbol{r}_L$. Therefore the total number of diagrams in this case is

$$\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L}) \times \sum_{\substack{1\le I\le K\\ p_I\neq r_J}}\sum_{1\le J\le L}\mathcal{N}_{g-1,k-1,l+2}(\boldsymbol{e_{2k+l}};\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)). \tag{41}$$

On the other hand, there are k chords to choose for removal in the original diagram, so the two expressions (40) and (41) must add up to $k\mathcal{N}_{g,k,l}(\boldsymbol{e}_{2k+l};\boldsymbol{q})$. □

The corresponding result for the multi-backbone case is the following.

Theorem 4.2. *$\mathcal{N}_{g,k,l}(\boldsymbol{b};\boldsymbol{q})$ satisfies the following recursion relation;*

$$\begin{aligned}
k\mathcal{N}_{g,k,l}(\boldsymbol{b};\boldsymbol{q}) &= \sum_{K\ge 1}\sum_{\boldsymbol{p}_K}(q_{\boldsymbol{p}_K}+1)\sum_{\substack{1\le I\le K\\ p_I\neq p_J}}\sum_{I<J\le K}\mathcal{N}_{g,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+s_{I,J}(\boldsymbol{p}_K))\\
&+\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L})\\
&\quad\sum_{\substack{1\le I\le K\\ p_I\neq r_J}}\sum_{1\le J\le L}\mathcal{N}_{g-1,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L))\\
&+\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}\sum_{g_1+g_2=g}\sum_{k_1+k_2=k-1}\sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}}\sum_{\substack{1\le I\le K\\ p_I\neq r_J}}\sum_{1\le J\le L}\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}}\\
&\quad q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{N}_{g_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{N}_{g_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)}),
\end{aligned}$$

where $b^{(a)}=\sum_{i=1}^{\infty}b_i^{(a)}$, $a=1,2$ and $s_{I,J}(\boldsymbol{p}_K)$ and $t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)$ are defined as in theorem 4.1.

Proof. The proof proceeds similarly to theorem 4.1, but we have to consider different cases when removing a chord from a diagram of type $\{g,k,l;\boldsymbol{b};\boldsymbol{q}\}$; after the removal of a chord, the diagram may be either connected or disconnected. In the first case, there are two subcases to consider; the chord to be removed may be adjacent to either two or one boundary component(s). Note in the second case, the chord to be removed must be adjacent to a single boundary component, since otherwise the two boundary components adjacent to the removed chord merge after the removal, which is absurd. So there are no subcases to consider in the second case.

The first case is exactly the same as theorem 4.1, so let us consider the second case. Suppose the resulting two connected components are of the type $\{g_1,k_1,l_1;\boldsymbol{b}(1);\boldsymbol{q}^{(1)}\}$ and $\{g_2,k_2,l_2;\boldsymbol{b}(2);\boldsymbol{q}^{(2)}\}$, with $k_1+k_2=k-1$, $l_1+l_2=l+2$, $g_1+g_2=g$, and $\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}$. Suppose also, that the two new boundary

components created as a result of the chord removal have the signatures $\boldsymbol{p}_K$ and $\boldsymbol{r}_L$. Then signature of the original boundary component prior to the removal is

$$\tilde{\boldsymbol{p}} = (p_1 \cdots p_{I-1} r_{J+1} r_{J+2} \cdots r_L r_1 r_2 \cdots r_{J-1} p_{I+1} p_{I+2} \cdots p_K),$$

with $1 \le I \le K$, $1 \le J \le L$, and $p_I \neq r_J$. If the point and type spectrum prior to the chord removal was $\boldsymbol{q}$, the sum of the spectra $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ is given by

$$\boldsymbol{q} - \boldsymbol{e}_{\tilde{\boldsymbol{p}}} + \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} = \boldsymbol{q} + t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L).$$

When adding a marked chord to obtain a diagram of type $\{g, k, l; \boldsymbol{b}; \boldsymbol{q}\}$, there are $q^{(1)}_{\boldsymbol{p}_K}$ boundary components to choose from the component $\boldsymbol{b}^{(1)}$, and $q^{(2)}_{\boldsymbol{r}_L}$ from $\boldsymbol{b}^{(2)}$. So the number of diagrams in this case is

$$\begin{aligned}
&\frac{1}{2} \sum_{K=1}^{\infty} \sum_{L=1}^{\infty} \sum_{\boldsymbol{p}_K} \sum_{\boldsymbol{r}_L} \sum_{g_1+g_2=g} \sum_{k_1+k_2=k-1} \sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}} \\
&\quad \times \sum_{\substack{1 \le I \le K \\ p_I \neq r_J}} \sum_{1 \le J \le L} \sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)} \\ =\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L)}} q^{(1)}_{\boldsymbol{p}_K} q^{(2)}_{\boldsymbol{r}_L} \frac{b!}{b^{(1)}! b^{(2)}!} \\
&\quad \times \mathcal{N}_{g_1,k_1,l_1}(\boldsymbol{b}^{(1)}; \boldsymbol{q}^{(1)}) \mathcal{N}_{g_2,k_2,l_2}(\boldsymbol{b}^{(2)}; \boldsymbol{q}^{(2)}),
\end{aligned}$$

with the factor $\frac{1}{2}$ accounting for the permutation of the two connected components, and $\frac{b!}{b^{(1)}! b^{(2)}!}$ for the ordered splitting of b backbone components. Together with the two terms from theorem 4.1, we obtain the claimed recursion.

□

Recursion relation for non-oriented protein diagrams

Non-oriented diagrams can be considered in the same way as in the RNA model, by assigning to each chord an extra binary datum to indicate whether it is twisted or not.

Let $\mathcal{M}_{h,k,l}(\boldsymbol{b}; \boldsymbol{q})$ be the number of both orientable and non-orientable diagrams of type $\{h, k, l; \boldsymbol{b}; \boldsymbol{q}\}$, where h is twice the genus in the orientable case and the number of cross-caps in the non-orientable case (see section 3.2). We start by considering one-backbone case.

Theorem 4.3. *$\mathcal{M}_{h,k,l}(\boldsymbol{e}_{2k+l}; \boldsymbol{q})$ satisfies the following relation.*

$$\begin{aligned}
k\mathcal{M}_{h,k,l}(\boldsymbol{e}_{2k+l}; \boldsymbol{q}) = &\\
&\sum_{K \ge 1} \sum_{\boldsymbol{p}_K} (q_{\boldsymbol{p}_K} + 1) \sum_{\substack{1 \le I \le K \\ p_I \neq p_J}} \sum_{I < J \le K} \big[2\mathcal{M}_{h,k-1,l+2}(\boldsymbol{e}_{2k+l}; \boldsymbol{q} + s_{I,J}(\boldsymbol{p}_K)) \\
&\qquad + \mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{e}_{2k+l}; \boldsymbol{q} + u_{I,J}(\boldsymbol{p}_K)) \big] \\
&+ \frac{1}{2} \sum_{K=1}^{\infty} \sum_{L=1}^{\infty} \sum_{\boldsymbol{p}_K} \sum_{\boldsymbol{r}_L} (q_{\boldsymbol{p}_K} + 1)(q_{\boldsymbol{r}_L} + 1 + \delta_{\boldsymbol{p}_K, \boldsymbol{r}_L}) \\
&\times \sum_{\substack{1 \le I \le K \\ p_I \neq r_J}} \sum_{1 \le J \le L} \big[\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{e}_{\boldsymbol{2k+l}}; \boldsymbol{q} + t_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L)) \\
&\qquad + \mathcal{M}_{h-2,k-1,l+2}(\boldsymbol{e}_{2k+l}; \boldsymbol{q} + \tilde{t}_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L)) \big],
\end{aligned}$$

where $s_{I,J}$ and $t_{I,J}$ are given in theorem 4.1, and

$$u_{I,J}(\boldsymbol{p}_K) = \boldsymbol{e}_{\boldsymbol{p}_K} - \boldsymbol{e}_{(p_1\cdots p_{I-1}p_{J-1}p_{J-2}\cdots p_{I+1}p_{J+1}\cdots p_K)},$$
$$\tilde{t}_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L) = \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} - \boldsymbol{e}_{(p_1\ldots p_{I-1}r_{J-1}\ldots r_1 r_L \ldots r_{J+1}p_{I+1}\ldots p_K)}.$$

Proof. When removing a chord, the removed chord is either twisted or not twisted. If it is not twisted we get the same expression as in theorem 4.1. So suppose the removed chord is twisted. We need to consider two cases; the removed chord is adjacent to two boundary components, or it is adjacent to just one. Let us start with the first case. The two boundary components adjacent to the removed chord are merged after the removal. This is clear from figure 25. So we have, after the chord removal, $n-1$ boundary components, $k-1$ chords,

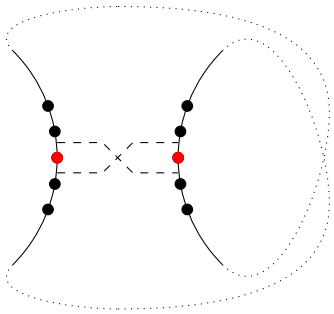


Figure 25: Removing a twisted chord where it is adjacent to two boundary components. The points represent marked points (the red points become marked points after the chord removal), and the dashed twisted band is the (twisted) chord to be removed. Observe the two boundary components are merged to one after the chord removal.

$l+2$ marked points, and the non-orientable genus h remain unchanged. Suppose the signature of the merged boundary is $\boldsymbol{p}_K = (p_1 \cdots p_K)$. Then prior to the removal, the two boundary components have the signatures

$$\boldsymbol{p}_1 = (p_1 p_2 \cdots p_{I-1} p_{J+1} p_{J+2} \cdots p_K),$$
$$\boldsymbol{p}_2 = (p_{I+1} p_{I+2} \cdots p_{J-1})$$

for $1 \leq I < J \leq K$ with $p_I \neq p_J$. We see that this case is the same as the removal of non-twisted chord that is adjacent to two boundary components, in theorem 4.1.

In the second case where the removed chord is adjacent to one boundary component, we have two subcases, one where the boundary component becomes split after the chord removal, and the other where the boundary component remains as a single component. The first subcase is illustrated in figure 26. If the two components after the removal have signatures $\boldsymbol{p}_K$ and $\boldsymbol{r}_L$, then prior to the removal the single boundary component has the signature

$$\tilde{\boldsymbol{p}} = (p_1 p_2 \cdots p_{I-1} r_{J-1} r_{J-2} \cdots r_1 r_L r_{L-1} \cdots r_{J+1} p_{I+1} p_{I+2} \cdots p_K),$$

with $1 \leq I \leq K$, $1 \leq J \leq L$ and $p_I \neq r_J$. if the spectrum prior to the removal was $\boldsymbol{q}$, then after the removal it becomes

$$\boldsymbol{q} - \boldsymbol{e}_{\tilde{\boldsymbol{p}}} + \boldsymbol{e}_{\boldsymbol{p}_K} + \boldsymbol{e}_{\boldsymbol{r}_L} = \boldsymbol{q} + \tilde{t}_{I,J}(\boldsymbol{p}_K, \boldsymbol{r}_L).$$

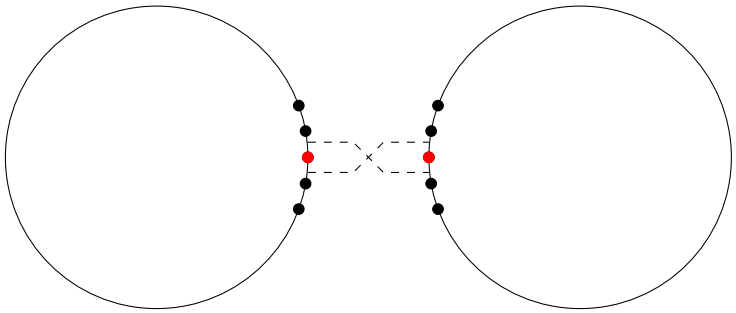


Figure 26: Removing a twisted chord where it is adjacent to a single boundary component, and the component splits after the removal. Note the orientation of the boundary before and after the removal.

There is a choice of $(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1)$ boundary components with signatures $\boldsymbol{p}_K$ and $\boldsymbol{r}_L$, if $\boldsymbol{p}_K \neq \boldsymbol{r}_L$, and $(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{p}_K}+2)/2$ is $\boldsymbol{p}_K = \boldsymbol{r}_L$. So in this case the number of diagrams is

$$\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L}) \times \sum_{\substack{1\le I\le K\\ p_I\neq r_J}}\sum_{1\le J\le L}\mathcal{M}_{h-2,k-1,l+2}(\boldsymbol{e}_{2k+l};\boldsymbol{q}+\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)).$$

The second subcase is illustrated in figure 27 After the removal, we have n

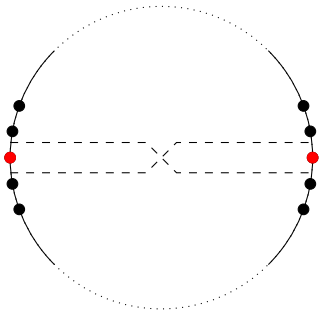


Figure 27: Removing a twisted chord where it is adjacent to a single boundary component, and the component remains connected after the removal. Note the orientation of the boundary before and after the removal.

boundary components, $k-1$ chords, $l+2$ marked points and $h-1$. If, after the removal, the boundary component has signature $\boldsymbol{p}_K = (p_1 \cdots p_K)$, the prior to removal the signature is

$$\tilde{\boldsymbol{p}} = (p_1 p_2 \cdots p_{I-1} p_{J-1} p_{J-2} \cdots p_{I+1} p_{J+1} p_{J+2} \cdots p_K),$$

with $1 \le I < J \le K$ and $p_I \neq p_J$. So if the spectrum prior to removal was give

by $\boldsymbol{q}$, then after the removal it becomes

$$\boldsymbol{q} - \boldsymbol{e}_{\tilde{\boldsymbol{p}}} + \boldsymbol{e}_{\boldsymbol{p}_K} = \boldsymbol{q} + u_{I,J}(\boldsymbol{p}_K).$$

There is a choice of $(q_{\boldsymbol{p}_K} + 1)$ boundary components with signature $\boldsymbol{p}_K$, so the number in this case is

$$\sum_{K\geq 1}\sum_{\boldsymbol{p}_K}(q_{\boldsymbol{p}_K}+1)\sum_{\substack{1\leq I\leq K\\ p_I\neq p_J}}\sum_{I<J\leq K}\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{e}_{2k+l};\boldsymbol{q}+u_{I,J}(\boldsymbol{p}_K)).$$

Adding all three terms to the r.h.s. of theorem 4.1, we obtain the claimed relation. □

The multi-backbone case is as follows.

Theorem 4.4. *$\mathcal{M}_{h,k,l}(\boldsymbol{b};\boldsymbol{q})$ satisfies the following relation.*

$$\begin{aligned}
k\mathcal{M}_{h,k,l}(\boldsymbol{b};\boldsymbol{q}) =\ & \sum_{K\geq 1}\sum_{\boldsymbol{p}_K}(q_{\boldsymbol{p}_K}+1)\sum_{\substack{1\leq I\leq K\\ p_I\neq p_J}}\sum_{I<J\leq K}\big[2\mathcal{M}_{h,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+s_{I,J}(\boldsymbol{p}_K))\\
& \quad +\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+u_{I,J}(\boldsymbol{p}_K))\big]\\
& +\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}(q_{\boldsymbol{p}_K}+1)(q_{\boldsymbol{r}_L}+1+\delta_{\boldsymbol{p}_K,\boldsymbol{r}_L})\\
& \times\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L}\big[\mathcal{M}_{h-1,k-1,l+2}(\boldsymbol{e_{2k+l}};\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L))\\
& \quad +\mathcal{M}_{h-2,k-1,l+2}(\boldsymbol{b};\boldsymbol{q}+\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L))\big]\\
& +\frac{1}{2}\sum_{K=1}^{\infty}\sum_{L=1}^{\infty}\sum_{\boldsymbol{p}_K}\sum_{\boldsymbol{r}_L}\sum_{h_1+h_2=h}\sum_{k_1+k_2=k-1}\sum_{\boldsymbol{b}^{(1)}+\boldsymbol{b}^{(2)}=\boldsymbol{b}}\sum_{\substack{1\leq I\leq K\\ p_I\neq r_J}}\sum_{1\leq J\leq L}\\
& \Bigg[\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+t_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}} q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{M}_{h_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{M}_{h_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)})\\
& +\sum_{\substack{\boldsymbol{q}^{(1)}+\boldsymbol{q}^{(2)}\\ =\boldsymbol{q}+\tilde{t}_{I,J}(\boldsymbol{p}_K,\boldsymbol{r}_L)}} q^{(1)}_{\boldsymbol{p}_K}q^{(2)}_{\boldsymbol{r}_L}\frac{b!}{b^{(1)}!b^{(2)}!}\mathcal{M}_{h_1,k_1,l_1}(\boldsymbol{b}^{(1)};\boldsymbol{q}^{(1)})\mathcal{M}_{h_2,k_2,l_2}(\boldsymbol{b}^{(2)};\boldsymbol{q}^{(2)})\Bigg],
\end{aligned}$$

where $s_{I,J}$ and $t_{I,J}$ are given in theorem 4.1, and $u_{I,J}$ and $\tilde{t}_{I,J}$ are given in theorem 4.3.

Proof. In addition to the one-backbone case, we need to consider the case where the removal of a twisted chord makes the diagram to be disconnected. If we concentrate on the boundary component to which the removed chord is part of, we have the situation illustrated in figure 26. So the same argument applies in the current case, and the claimed relation follows. □

4.4 The protein matrix model

We now present a matrix model for protein diagrams. The basic building block is the model of peptide units, as described in section 4.2. For the purpose of this section (and in accordance with the presentation in section 3.3), we draw this as a surface by thickening edges (figure 28).

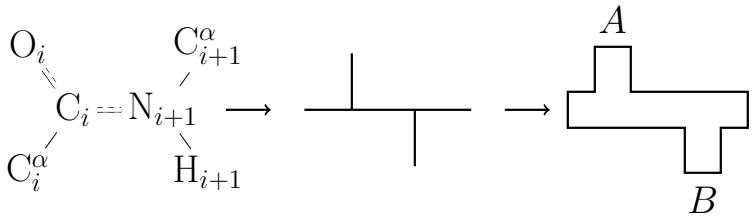


Figure 28: Peptide unit and its fatgraph representation.

We attach matrices A and B to the half-edges representing the carboxyl oxygen and amino hydrogen, respectively. The backbone is constructed by connecting two or more peptide units together at the half-edges representing the α-carbon atoms. There are two ways to connect peptide units; twisted or not twisted. To start wih, we will not allow mixing of twisted and untwisted connections on a single backbone, but we will relax this requirement later. We present the two backbone configurations in figure 29. These two configurations correspond to the two typical secondary structures, α-helix and β-sheet. In the α-helix configuration, the peptide units are connected without any twists, while in the β-sheet configuration, all connections between peptide units are twisted. In the language of the matrix model, each backbone structure gives rise to a vertex. We assign the trace $\mathrm{Tr}A^iB^i$ to the α-helix backbone with i peptide units, and the trace $\mathrm{Tr}A^iB^i$ to the β-sheet backbone with i peptide units.

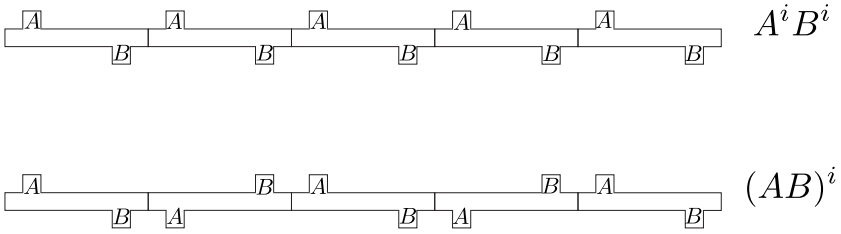


Figure 29: Two types of backbones; α-helix (above) and β-sheet (below).

The hydrogen bonds are represented as the bands, also called the propagators in the matrix model, which connect A and B. To represent unpaired H's and N's, univalent vertices are introduced to cap A and B ends. The univalent vertex replaces the matrix A by the external matrix Λ_1, and the matrix B by the external matrix Λ_2. In the fatgraph diagram, external matrices Λ_1 and Λ_2 are represented by marked points with different colors. We construct the protein fatgraphs by decorating one or more α- and β-backbones with bands and marked points. (See figures 30 to 31.)

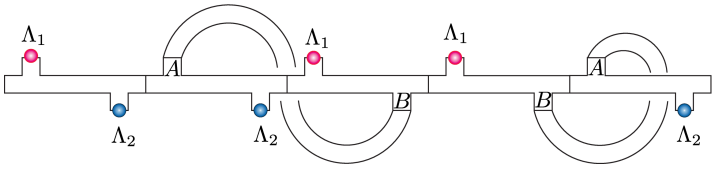


Figure 30: A fatgraph model for α-helix.

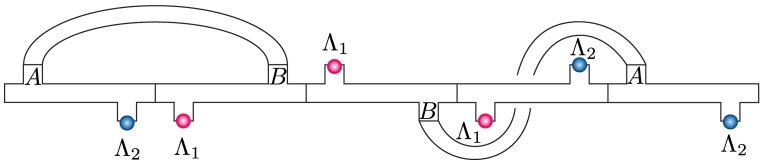


Figure 31: A fatgraph model for β-sheet.

For each boundary component in the fatgraph, we have a (possibly empty) sequence of unpaired hydrogens and oxygens, represented by the external matrices Λ_1 and Λ_2. To record these sequences, we introduce a combinatorial parameter, which we call the *boundary point type spectrum*, $\boldsymbol{\ell_{ij}} = \{\ell_{\boldsymbol{ij}}\}$. It is a sequence of numbers $\ell_{\boldsymbol{ij}}$, indexed by two sequences $\boldsymbol{i} = (i_1, \ldots, i_K)$ and $\boldsymbol{j} = (j_1, \ldots, j_K)$. For each boundary component, $\boldsymbol{i}$ records the numbers of consecutive unpaired oxygens, and $\boldsymbol{j}$ records the numbers of consecutive hydrogens. So for example, the diagram in figure 30 has the boundary point type spectrum $(\ell_{(1),(2)} = 1, \ell_{(2),(0)} = 1, \ell_{(0),(1)} = 1)$, while the diagram in figure 31 has the spectrum $(\ell_{(0),(0)} = 1, \ell_{(1,1),(1,1)} = 1, \ell_{(1),(1)} = 1)$. The total number $2l$ of unpaired hydrogens and oxygens is given by

$$\begin{aligned} l &= \sum_{K\geq 1} \sum_{(i_1,\ldots,i_K)} \sum_{(j_1,\ldots,j_K)} \sum_{L=1}^{K} i_L \ell_{(i_1,\ldots,i_K)(j_1,\ldots,j_K)} \\ &= \sum_{K\geq 1} \sum_{(i_1,\ldots,i_K)} \sum_{(j_1,\ldots,j_K)} \sum_{L=1}^{K} j_L \ell_{(i_1,\ldots,i_K)(j_1,\ldots,j_K)}. \end{aligned}$$

We also require two backbone spectra, $\boldsymbol{a} = \{a_i\}$ and $\boldsymbol{b} = \{b_i\}$, for the numbers of backbone segments of each type with i peptide units. For the diagram in figure 30, we have $\{a_i\} = \boldsymbol{e}_5, \{b_i\} = 0$, and for figure 31 we have $\{a_i\} = 0, \{b_i\} = \boldsymbol{e}_5$.

Let $a = \sum_{i\geq 1}$ be the total number of peptide units in the α-helix backbones, and $b = \sum_{i\geq 1}$ be the total number in the β-sheet backbones.

Definition 4. Let $\mathcal{N}_{g,k,l}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\ell_{ij}})$ denote the number of protein fatgraphs with genus g, k propagators, $2l$ marked points, $\boldsymbol{a}$ backbone spectrum for the untwisted backbones (i.e. α-helix), $\boldsymbol{b}$ backbone spectrum for the twisted backbones (i.e. β-sheet), and $\boldsymbol{\ell_{ij}}$ boundary point spectrum. The generating function of the

number $\mathcal{N}_{g,k,l}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\ell_{ij}})$ is defined by

$$\begin{aligned}
&F(x,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) = \sum_{b\geq 0} F_{a,b}(x,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}),\\
&F_{a,b}(x,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&= \frac{1}{a!b!}\sum_{g\geq 0}\sum_{k\geq a+b-1}\sum_{\boldsymbol{\ell_{ij}}}\sum_{\sum a_i=a}\sum_{\sum b_i=b} \mathcal{N}_{g,k,l}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\ell_{ij}})x^{2g-2}y^k\\
&\quad\times\prod_{i\geq 0}\alpha_i^{a_i}\beta_i^{b_i}\prod_{\boldsymbol{ij}} r_{\boldsymbol{ij}}^{\ell_{\boldsymbol{ij}}}.
\end{aligned}$$

Using Wick's theorem with the Wick conrtaction

$$\langle A_{ab}B_{cd}\rangle = \frac{1}{\mathrm{Vol}(\mathcal{H}_N)^2}\int_{\mathcal{H}_N} dAdB\ A_{ab}B_{cd}\mathrm{e}^{-\mathrm{Tr}AB} = \delta_{ad}\delta_{bc}, \tag{42}$$

we can express the generating function as a hermitian matrix integral.

Theorem 4.5. *Let $Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ denote the exponential of the generating function:*

$$Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) = \exp\left[F(1/N,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\right].$$

$Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ is given as the partition function of the hermitian 2-matrix model with external fields Λ_1 and Λ_2:

$$\begin{aligned}
&Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&= \frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}} dAdB\ \exp\Bigg[-N\mathrm{Tr}\Bigg(AB - \sum_{i\geq 0}\alpha_i y^i (A+y^{-1/2}\Lambda_1)^i(B+y^{-1/2}\Lambda_2)^i\\
&\qquad\qquad - \sum_{i\geq 0}\beta_i y^i((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^i\Bigg)\Bigg]\\
&= \frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}} dAdB\ \mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)},
\end{aligned}\tag{43}$$

where $\mathrm{Vol}_N = N^{N(N+1)/2}\mathrm{Vol}(\mathcal{H}_N)$, and $r_{\boldsymbol{ij}}$'s are defined by the single trace for a product of Λ_1's and Λ_2's as

$$r_{(i_1,\ldots,i_K),(j_1,\ldots,j_K)} = \frac{1}{N}\mathrm{Tr}(\Lambda_1^{i_1}\Lambda_2^{j_1}\Lambda_1^{i_2}\Lambda_2^{j_2}\cdots\Lambda_1^{i_K}\Lambda_2^{j_K}).$$

Proof. The construction is done similarly to section 3.3, by assigning the appropriate elements to the diagram elements as described above. □

This generating function obeys the heat equation.

Theorem 4.6. *The generating function $Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ satisfies the heat equation:*

$$\begin{aligned}
&\frac{\partial}{\partial y}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&= \frac{1}{2N}\left(\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2} + \mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}),
\end{aligned}\tag{44}$$

where $\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}$ *denotes*

$$\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}=\sum_{a,b=1}^{N}\frac{\partial^2}{\partial\Lambda_{1ab}\partial\Lambda_{2ba}}.$$

Proof. The heat equation for the partition function $Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})$ is obtained by the shift invariance of the matrix integral measure dA and dB.

$$\begin{aligned}
&\frac{\partial}{\partial\Lambda_{1ba}}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}}dAdB\ N\sum_{i\geq 1}y^{i-1/2}\\
&\quad\times\Bigg(\alpha_i\sum_{k=0}^{i-1}\big((A+y^{-1/2}\Lambda_1)^k(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^{i-k-1}\big)_{ab}\\
&\qquad+i\beta_i\big((B+y^{-1/2}\Lambda_2)((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{i-1}\big)_{ab}\Bigg)\\
&\quad\times\mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)}\\
&=\frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}}dXdY\ N\sum_{i\geq 1}y^{i-1/2}\left(\alpha_i\sum_{k=0}^{i-1}(X^kY^iX^{i-k-1})+i\beta_iY(XY)^{i-1}\right)_{ab}\\
&\quad\times\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y;\Lambda_1,\Lambda_2)},
\end{aligned}$$

where $X=A+y^{-1/2}\Lambda_1$, $Y=B+y^{-1/2}\Lambda_2$, and

$$\begin{aligned}
&W_{y,\alpha,\beta}(X,Y;\Lambda_1,\Lambda_2)\\
&=(X-y^{-1/2}\Lambda_1)(Y-y^{-1/2}\Lambda_2)-\sum_{i\geq 0}\alpha_iy^iX^iY^i-\sum_{i\geq 0}\beta_iy^i(XY)^i.
\end{aligned}$$

We then compute the derivative $\sum_{a,b=1}^{N}\partial/\partial\Lambda_{2ab}$ to find

$$\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}}dXdY\ N^2\sum_{i\geq 1}y^{i-1}\\
&\qquad\times\mathrm{Tr}\left(\left(\alpha_i\sum_{k=0}^{i-1}X^kY^iX^{i-k-1}+i\beta_iY(XY)^{i-1}\right)(X-y^{-1/2}\Lambda_1)\right)\\
&\qquad\times\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y;\Lambda_1,\Lambda_2)}.
\end{aligned}$$

Exchanging the role of (X, Λ_1) and (Y, Λ_2), we find

$$\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}}dXdY\ N^2\sum_{i\geq 1}y^{i-1}\\
&\qquad\times\mathrm{Tr}\left(\left(\alpha_i\sum_{k=0}^{i-1}Y^kX^iY^{i-k-1}+i\beta_i(XY)^{i-1}X\right)(Y-y^{-1/2}\Lambda_2)\right)\\
&\qquad\times\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y;\Lambda_1,\Lambda_2)}. \qquad (45)
\end{aligned}$$

On the other hand, the derivative with respect to y is

$$\begin{aligned}
&\frac{\partial}{\partial y}Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^2}\int_{\mathcal{H}_N^{\otimes 2}}dAdB\ \frac{N}{2}\sum_{i\geq 1}y^{i-1}\\
&\quad\times\mathrm{Tr}\Bigg[\alpha_i\sum_{k=0}^{i-1}\Big((A+y^{-1/2}\Lambda_1)^kA(A+y^{-1/2}\Lambda_1)^{i-k-1}(B+y^{-1/2}\Lambda_2)^i\\
&\qquad\qquad+(B+y^{-1/2}\Lambda_2)^kB(B+y^{-1/2}\Lambda_2)^{i-k-1}(A+y^{-1/2}\Lambda_1)^i\Big)\\
&\quad+i\beta_i\Big(A(B+y^{-1/2}\Lambda_2)+(A+y^{-1/2}\Lambda_1)B\Big)\Big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\Big)^{i-1}\Bigg]\\
&\quad\times\mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)}\\
&=\frac{1}{2N}\left(\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}).
\end{aligned}$$

□

The initial condition for this heat equation is found by setting $y = 0$ in (43).

$$Z_N(y=0;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})=\mathrm{e}^{N\sum_{i\geq 0}\mathrm{Tr}(\alpha_i\Lambda_1^i\Lambda_2^i+\beta_i(\Lambda_1\Lambda_2)^i)}.$$

The above heat equation can be expressed as a cut-and-join equation.

Theorem 4.7. *Let L_0 and L_2 denote the following differential operators with*

respect to parameters r_{ij}*;*

$$L_0 = \sum_{K\geq 1} \sum_{\{i_L,j_L\}_{L=1}^K} \sum_{L=1}^K \sum_{M=1}^K \sum_{k=1}^{i_L} \sum_{\ell=1}^{j_M} r_{(i_L-k-1,i_{L+1},\dots,i_M),(j_L,\dots,j_{M-1},\ell)} r_{(k,i_{M+1},\dots,i_{L-1}),(j_M-\ell-1,j_{M+1},\dots,j_{L-1})} \times \frac{\partial}{\partial r_{(i_1,\dots,i_K),(j_1,\dots,j_K)}},$$

$$L_2 = \sum_{K,V\geq 1} \sum_{\{i_L,j_L\}_{L=1}^K} \sum_{\{s_Q,t_Q\}_{Q=1}^V} \sum_{L=1}^K \sum_{Q=1}^V \sum_{k=1}^{i_L} \sum_{u=1}^{t_Q} r_{(i_L-k-1,i_{L+1},\dots,i_{L-1},k,s_{Q+1},\dots,s_{Q-1},s_Q),(j_L,j_{L+1},\dots,j_{L-1},t_Q-u-1,t_{Q+1},\dots,t_{Q-1},u)} \times \frac{\partial^2}{\partial r_{(i_1,\dots,i_K),(j_1,\dots,j_K)} \partial r_{(s_1,\dots,s_V),(t_1,\dots,t_V)}},$$

where the labels L, M *are defined modulo* K*, and the label* Q *is defined modulo* V*.*

The heat equation (44) is rewritten as the cut-and-join equation:

$$\frac{\partial Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})}{\partial y} = \left(L_0 + \frac{1}{N^2} L_2\right) Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}).$$

Proof. By the chain rule applied to the right hand side of the heat equation (44), we find

$$\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2} Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}}) \\
&= \sum_{K\geq 1} \sum_{\{i_L,j_L\}_{L=1}^K} \mathrm{Tr}\frac{\partial^2 r_{(i_1,\dots,i_K),(j_1,\dots,j_K)}}{\partial\Lambda_1\partial\Lambda_2} \frac{\partial Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})}{\partial r_{(i_1,\dots,i_K),(j_1,\dots,j_K)}} \\
&\quad + \sum_{K,V\geq 1} \sum_{\{i_L,j_L\}_{L=1}^K} \sum_{\{s_Q,t_Q\}_{Q=1}^V} \mathrm{Tr}\frac{\partial r_{(i_1,\dots,i_K),(j_1,\dots,j_K)}}{\partial\Lambda_1} \frac{\partial r_{(s_1,\dots,s_V),(t_1,\dots,t_V)}}{\partial\Lambda_2} \\
&\qquad\qquad \times \frac{\partial^2 Z_N(y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{r_{ij}})}{\partial r_{(i_1,\dots,i_K),(j_1,\dots,j_K)} \partial r_{(s_1,\dots,s_V),(t_1,\dots,t_V)}}. \quad (46)
\end{aligned}$$

The coefficients in (46) are;

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2 r_{(i_1,\ldots,i_K),(j_1,\ldots,j_K)}}{\partial\Lambda_1\partial\Lambda_2}\\
&=\frac{1}{N}\sum_{L,M=1}^{K}\sum_{k=1}^{i_L}\sum_{\ell=1}^{j_M}\mathrm{Tr}(\Lambda_1^{i_L-k-1}\Lambda_2^{j_L}\cdots\Lambda_1^{i_M}\Lambda_2^{\ell})\\
&\qquad\qquad\times\mathrm{Tr}(\Lambda_2^{j_M-\ell-1}\Lambda_1^{i_{M+1}}\Lambda_2^{j_{M+1}}\cdots\Lambda_1^{i_{L-1}}\Lambda_2^{j_{L-1}}\Lambda_1^{k})\\
&=N\sum_{L,M=1}^{K}\sum_{k=1}^{i_L}\sum_{\ell=1}^{j_M}r_{(i_L-k-1,i_{L+1},\ldots,i_M),(j_L,\ldots,j_{M-1},\ell)}\\
&\qquad\qquad\times r_{(k,i_{M+1},\ldots,i_{L-1}),(j_M-\ell-1,j_{M+1},\ldots,j_{L-1})},\\
&\mathrm{Tr}\frac{\partial r_{(i_1,\ldots,i_K),(j_1,\ldots,j_K)}}{\partial\Lambda_1}\frac{\partial r_{(s_1,\ldots,s_V),(t_1,\ldots,t_V)}}{\partial\Lambda_2}\\
&=\frac{1}{N^2}\sum_{L=1}^{K}\sum_{Q=1}^{V}\sum_{k=1}^{i_L}\sum_{u=1}^{t_Q}\mathrm{Tr}(\Lambda_1^{i_L-k-1}\Lambda_2^{j_L}\Lambda_1^{i_L+1}\Lambda_2^{j_{L+1}}\cdots\Lambda_1^{i_{L-1}}\Lambda_2^{j_{L-1}}\Lambda_1^{k}\\
&\qquad\qquad\cdot\Lambda_2^{t_Q-u-1}\Lambda_1^{s_{Q+1}}\Lambda_2^{t_{Q+1}}\cdots\Lambda_1^{s_{Q-1}}\Lambda_2^{t_{Q-1}}\Lambda_1^{s_Q}\Lambda_2^{u})\\
&=\frac{1}{N}\sum_{L=1}^{K}\sum_{Q=1}^{V}\sum_{k=1}^{i_L}\sum_{u=1}^{t_Q}\\
&\qquad r_{(i_L-k-1,i_{L+1},\ldots,i_{L-1},k,s_{Q+1},\ldots,s_{Q-1},s_Q),(j_L,j_{L+1},\ldots,j_{L-1},t_Q-u-1,t_{Q+1},\ldots,t_{Q-1},u)}.
\end{aligned}
$$

Thus we find the operators L_0 and L_2. Summing with the results of $\mathrm{Tr}\frac{\partial^2}{\Lambda_2\Lambda_1}$ accounts for the factor $\frac{1}{2N}$ in (44).

□

Merging backbones

We will now slightly relax the initial requirement that a backbone can only contain one type of connection between the peptide units by introducing another matrix M. To the endpoints of backbones, we attach the matrix M, with propagators connecting these endpoints to create a loop structure (figure 32).

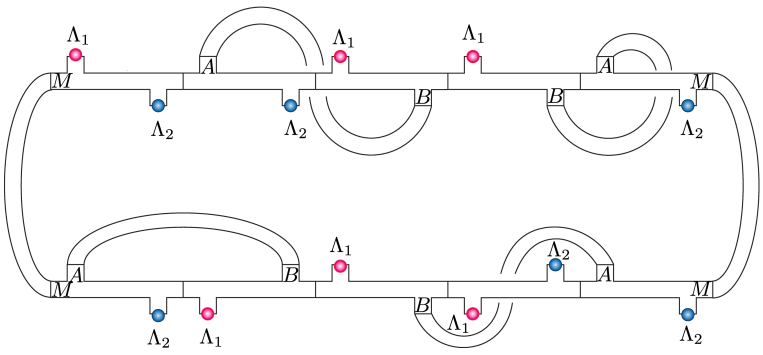


Figure 32: Connecting α-helix and β-sheet backbones.

Definition 5. The partition function of the protein matrix model for merged backbones is defined as the following hermitian 3-matrix model:

$$
\begin{aligned}
&Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N)^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\,\exp\Bigg[-N\mathrm{Tr}\Bigg(AB+\frac{1}{2}M^2\\
&\quad+\sum_{i\geq 0} y^i\Bigg\{\alpha_i^{(1)}M(A+y^{-1/2}\Lambda_1)^i(B+y^{-1/2}\Lambda_2)^iM\\
&\qquad\qquad+\alpha_i^{(2)}M(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^iM\\
&\qquad\qquad+\beta_i^{(1)}M\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^iM\\
&\qquad\qquad+\beta_i^{(2)}M\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^iM\Bigg\}\Bigg)\Bigg]\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N)^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\,\mathrm{e}^{-N\mathrm{Tr}V_{\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2)}.
\end{aligned}
$$

The propagator $\langle A_{ab}B_{cd}\rangle$ of this matrix model represents the hydrogen bondings and the propagator $\langle M_{ab}M_{cd}\rangle$ represents the the loop that connects the α-helices and β-sheets in the backbones.

The heat equation for the model is as follows.

Theorem 4.8. *The partition function* $Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})$ *obeys the heat equation,*

$$
\begin{aligned}
&\frac{\partial Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})}{\partial y}\\
&=\frac{1}{2N}\mathrm{Tr}\left(\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}}).
\end{aligned}
$$

Proof. First, we consider the derivative $\partial/\partial\Lambda_1$ of the partition function $Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})$.

$$
\begin{aligned}
&\frac{\partial}{\partial\Lambda_{1ba}}Z_N(y;\{\alpha_i^{(1)}\},\{\alpha_i^{(2)}\},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}}dAdBdM\ N\sum_{i\geq 1}y^{i-1/2}\\
&\quad\times\Bigg(\alpha_i^{(1)}\sum_{k=0}^{i-1}\big((A+y^{-1/2}\Lambda_1)^k(B+y^{-1/2}\Lambda_2)^iM^2(A+y^{-1/2}\Lambda_1)^{i-k-1}\big)_{ab}\\
&\qquad+\alpha_i^{(2)}\sum_{k=0}^{i-1}\big((A+y^{-1/2}\Lambda_1)^kM^2(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^{i-k-1}\big)_{ab}\\
&\qquad+\beta_i^{(1)}\sum_{k=0}^{i-1}\big((B+y^{-1/2}\Lambda_2)((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^kM^2\\
&\qquad\qquad\times((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2))^{i-k-1}\big)_{ab}\\
&\qquad+\beta_i^{(2)}\sum_{k=0}^{i-1}\big(((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^kM^2\\
&\qquad\qquad\times((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1))^{i-k-1}(B+y^{-1/2}\Lambda_2)\big)_{ab}\Bigg)\\
&\quad\times\mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2)}\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}}dXdYdM\ N\sum_{i\geq 1}y^{i-1/2}\Bigg(\\
&\qquad+\alpha_i^{(1)}\sum_{k=0}^{i-1}(X^kY^iM^2X^{i-k-1})+\alpha_i^{(2)}\sum_{k=0}^{i-1}(X^kM^2Y^iX^{i-k-1})\\
&\qquad+\beta_i^{(1)}\sum_{k=0}^{i-1}Y(XY)^kM^2(XY)^{i-k-1}+\beta_i^{(2)}\sum_{k=0}^{i-1}(YX)^kM^2(YX)^{i-k-1}Y\Bigg)_{ab}\\
&\qquad\times\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)},
\end{aligned}
$$

where $X=A+y^{-1/2}\Lambda_1$, $Y=B+y^{-1/2}\Lambda_2$, and

$$
\begin{aligned}
&W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)\\
&=(X-y^{-1/2}\Lambda_1)(Y-y^{-1/2}\Lambda_2)+\frac{1}{2}M^2\\
&\quad-\sum_{i\geq 0}y^i(\alpha_i^{(1)}MX^iY^iM+\alpha_i^{(2)}MY^iX^iM)\\
&\quad-\sum_{i\geq 0}y^i(\beta_i^{(1)}M(XY)^iM+\beta_i^{(2)}M(YX)^iM).
\end{aligned}
$$

We now compute the derivative $\sum_{a,b=1}^{N} \partial/\partial\Lambda_{2ab}$ to find

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\{\beta_i^{(1)}\},\{\beta_i^{(2)}\},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}} dXdYdM\ N^2\sum_{i\geq 1}y^{i-1}\\
&\quad\times\mathrm{Tr}\Bigg(\Big(\alpha_i^{(1)}\sum_{k=0}^{i-1}X^kY^iM^2X^{i-k-1}+\alpha_i^{(2)}\sum_{k=0}^{i-1}X^kM^2Y^iX^{i-k-1}\\
&\qquad+\beta_i^{(1)}\sum_{k=0}^{i-1}Y(XY)^kM^2(XY)^{i-k-1}+\beta_i^{(2)}\sum_{k=0}^{i-1}(YX)^kM^2(YX)^{i-k-1}Y\Big)\\
&\qquad\times(X-y^{-1/2}\Lambda_1)\Bigg)\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)}\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}} dXdYdM\ N^2\sum_{i\geq 1}y^{i-1}\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)}\\
&\quad\times\mathrm{Tr}\Bigg(\Big(\alpha_i^{(1)}\sum_{k=1}^{i-1}MX^{i-k-1}(X-y^{-1/2}\Lambda_1)X^kY^iM\\
&\qquad+\alpha_i^{(2)}\sum_{k=1}^{i-1}MY^iX^{i-k-1}(X-y^{-1/2}\Lambda_1)X^kM\\
&\qquad+\beta_i^{(1)}\sum_{k=1}^{i-1}M(XY)^{i-k-1}(X-y^{-1/2}\Lambda_1)Y(XY)^kM\\
&\qquad+\beta_i^{(2)}\sum_{k=1}^{i-1}M(YX)^{i-k-1}Y(X-y^{-1/2}\Lambda_1)(YX)^kM\Big).
\end{aligned}
\tag{47}
$$

Exchanging the role of (X,Λ_1) and (Y,Λ_2), we find

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}Z_N(y;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\{\beta_i^{(1)}\},\{\beta_i^{(2)}\},\boldsymbol{r_{ij}})\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}} dXdYdM\ N^2\sum_{i\geq 1}y^{i-1}\mathrm{e}^{-N\mathrm{Tr}W_{y,\alpha,\beta}(X,Y,M;\Lambda_1,\Lambda_2)}\\
&\quad\times\mathrm{Tr}\Bigg(\Big(\alpha_i^{(1)}\sum_{k=1}^{i-1}MX^iy^{i-k-1}(Y-y^{-1/2}\Lambda_2)Y^kM\\
&\qquad+\alpha_i^{(2)}\sum_{k=1}^{i-1}MY^{i-k-1}(Y-y^{-1/2}\Lambda_2)Y^kX^iM\\
&\qquad+\beta_i^{(1)}\sum_{k=1}^{i-1}M(XY)^{i-k-1}X(Y-y^{-1/2}\Lambda_2)(XY)^kM\\
&\qquad+\beta_i^{(2)}\sum_{k=1}^{i-1}M(YX)^{i-k-1}(Y-y^{-1/2}\Lambda_2)X(YX)^kM\Big).
\end{aligned}
\tag{48}
$$

Finally, we compute the derivative with respect to y to find

$$
\begin{aligned}
&\frac{\partial}{\partial y} Z_N(y;\{\alpha_i\},\{\beta_i\},\boldsymbol{r_{ij}}) \\
&= \frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\ \frac{N}{2}\sum_{i\geq 1} y^{i-1} \\
&\quad\times \mathrm{Tr}\Bigg[\alpha_i^{(1)}\sum_{k=0}^{i-1}\Big(M(A+y^{-1/2}\Lambda_1)^k A(A+y^{-1/2}\Lambda_1)^{i-k-1}(B+y^{-1/2}\Lambda_2)^i M \\
&\qquad\qquad +M(A+y^{-1/2}\Lambda_1)^i(B+y^{-1/2}\Lambda_2)^i B(B+y^{-1/2}\Lambda_2)^{i-k-1}M\Big) \\
&\qquad +\alpha_i^{(2)}\sum_{k=0}^{i-1}\Big(M(B+y^{-1/2}\Lambda_2)^k B(B+y^{-1/2}\Lambda_2)^{i-k-1}(A+y^{-1/2}\Lambda_1)^i M \\
&\qquad\qquad +M(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^k A(A+y^{-1/2}\Lambda_1)^{i-k-1}\Big) \\
&\quad +\beta_i^{(1)}\sum_{k=0}^{i-1}\Big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\Big)^k \\
&\quad\times\Big(A(B+y^{-1/2}\Lambda_2)+(A+y^{-1/2}\Lambda_1)B\Big)\Big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\Big)^{i-k-1} \\
&\quad +\beta_i^{(2)}\sum_{k=0}^{i-1}\Big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\Big)^k \\
&\quad\times\Big(B(A+y^{-1/2}\Lambda_1)+(B+y^{-1/2}\Lambda_2)A\Big)\Big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\Big)^{i-k-1}\Bigg] \\
&\quad\times \mathrm{e}^{-N\mathrm{Tr}V_{y,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2)} \\
&= \frac{1}{2N}\left(\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\mathrm{Tr}\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right) Z_N(y;\{\alpha_i\},\{\beta_i\},\boldsymbol{r_{ij}}). \qquad (49)
\end{aligned}
$$

□

For the initial condition of the heat equation, we find

$$
\begin{aligned}
&Z_N(y=0;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij}}) \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N)}\int_{\mathcal{H}_N} dM\ \mathrm{e}^{-\frac{N}{2}\mathrm{Tr}M\left(I_N-2\sum_{i\geq 0}(\alpha_i^{(1)}\Lambda_1^i\Lambda_2^i+\alpha_i^{(2)}\Lambda_2^i\Lambda_1^i+\beta_i^{(1)}(\Lambda_1\Lambda_2)^i+\beta_i^{(2)}(\Lambda_2\Lambda_1)^i)\right)M} \\
&= \det\left(I_N-2\sum_{i\geq 0}(\alpha_i^{(1)}\Lambda_1^i\Lambda_2^i+\alpha_i^{(2)}\Lambda_2^i\Lambda_1^i+\beta_i^{(1)}(\Lambda_1\Lambda_2)^i+\beta_i^{(2)}(\Lambda_2\Lambda_1)^i)\right)^{-1/2} \\
&= \exp\left(\sum_{n=1}^{\infty}\frac{1}{n}\mathrm{Tr}\left(\sum_{i\geq 0}(\alpha_i^{(1)}\Lambda_1^i\Lambda_2^i+\alpha_i^{(2)}\Lambda_2^i\Lambda_1^i+\beta_i^{(1)}(\Lambda_1\Lambda_2)^i+\beta_i^{(2)}(\Lambda_2\Lambda_1)^i)\right)^n\right),
\end{aligned}
$$

where the Plemelj's formula is used

$$
\det(I_N+X)=\exp\left(\sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{n}\mathrm{Tr}X^n\right).
$$

Introducing N- and C-termini

We extend the protein matrix model further by introducing yet another external matrix Λ, which labels N- and C-termini of the backbones (figure 33 and figure 34). The boundary cycles containing p backbone ends are labelled by the set of numbers $(i_1^{(1)}, \ldots, i_{K_1}^{(1)} : \cdots : i_1^{(p)}, \ldots, i_{K_p}^{(p)})$ that count the number of unpaired carboxyl oxygens (Λ_1) and $(j_1^{(1)}, \ldots, j_{K_1}^{(1)} : \cdots : j_1^{(p)}, \ldots, j_{K_p}^{(p)})$ that count the number of unpaired amino hydrogens (Λ_2) keeping their ordering on the boundary cycle.

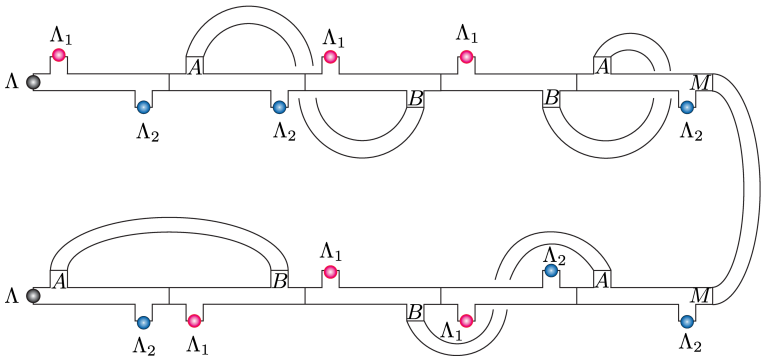


Figure 33: Adding C- and N-ends of backbone

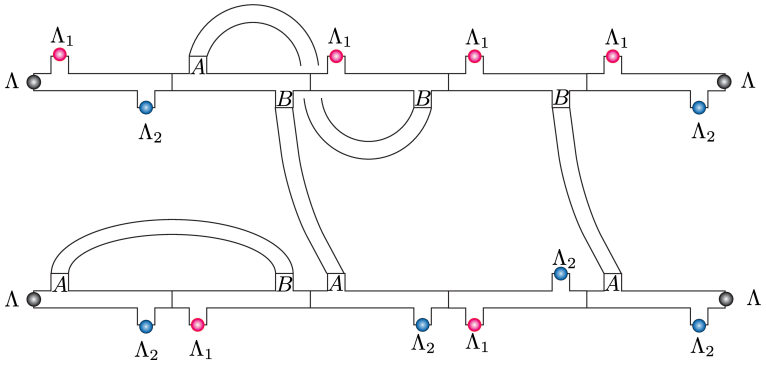


Figure 34: Two backbones with C- and N-ends

Let $\boldsymbol{n}_{\boldsymbol{ij},p}$ denote the extended boundary point type spectrum that counts the number $n_{\boldsymbol{ij};p}$ of boundary components containing a sequence of $\boldsymbol{i}$ Λ_1's a sequence of $\boldsymbol{j}$ Λ_2's, and p backbone end points.

Definition 6. The partition function of the protein matrix model with back-

bone endpoints is defined as the following hermitian 3-matrix model;

$$
\begin{aligned}
&Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{i,j;p}})\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N)^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\,\exp\Bigg[-N\mathrm{Tr}\Bigg(AB+\frac{1}{2}M^2\\
&\quad-\sum_{i\geq 0}y^i\eta(M+\eta^{-1/2}\Lambda)\Bigg\{\\
&\qquad+\alpha_i^{(1)}(A+y^{-1/2}\Lambda_1)^i(B+y^{-1/2}\Lambda_2)^i\\
&\qquad+\alpha_i^{(2)}(B+y^{-1/2}\Lambda_2)^i(A+y^{-1/2}\Lambda_1)^i\\
&\qquad+\beta_i^{(1)}\big((A+y^{-1/2}\Lambda_1)(B+y^{-1/2}\Lambda_2)\big)^i\\
&\qquad+\beta_i^{(2)}\big((B+y^{-1/2}\Lambda_2)(A+y^{-1/2}\Lambda_1)\big)^i\Bigg\}(M+\eta^{-1/2}\Lambda)\Bigg)\Bigg]\\
&=\frac{1}{\mathrm{Vol}(\mathcal{H}_N)^3}\int_{\mathcal{H}_N^{\otimes 3}} dAdBdM\,\mathrm{e}^{-N\mathrm{Tr}V_{y,\eta,\alpha,\beta}(A,B,M;\Lambda_1,\Lambda_2,\Lambda)}. \qquad (50)
\end{aligned}
$$

The parameter $r_{\boldsymbol{i},\boldsymbol{j};,p}$ is given by

$$
\begin{aligned}
&r_{\boldsymbol{i},\boldsymbol{j};,p}\\
&=r_{(i_1^{(1)},\ldots,i_{K_1}^{(1)}:i_1^{(2)},\ldots,i_{K_2}^{(2)}:\ldots:i_1^{(p)},\ldots,i_{K_p}^{(p)}),(j_1^{(1)},\ldots,j_{K_1}^{(1)}:j_1^{(2)},\ldots,j_{K_2}^{(2)}:\ldots:j_1^{(p)},\ldots,j_{K_p}^{(p)})}\\
&=\frac{1}{N}\mathrm{Tr}\Big(\Lambda_1^{i_1^{(1)}}\Lambda_2^{j_1^{(1)}}\cdots\Lambda_1^{i_{K_1}^{(1)}}\Lambda_2^{j_{K_1}^{(1)}}\Lambda\Lambda_1^{i_1^{(2)}}\Lambda_2^{j_1^{(2)}}\cdots\Lambda_1^{i_{K_2}^{(2)}}\Lambda_2^{j_{K_2}^{(2)}}\Lambda\cdots\Lambda_1^{i_1^{(p)}}\Lambda_2^{j_1^{(p)}}\cdots\Lambda_1^{i_{K_p}^{(p)}}\Lambda_2^{j_{K_p}^{(p)}}\Lambda\Big).
\end{aligned}
$$

The heat equations are as follows.

Theorem 4.9. *The partition function $Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})$ obeys heat equations:*

$$
\left(\frac{\partial}{\partial y}-\frac{1}{2N}\left(\frac{\partial^2}{\partial\Lambda_1\partial\Lambda_2}+\frac{\partial^2}{\partial\Lambda_2\partial\Lambda_1}\right)\right)Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})=0, \qquad (51)
$$

$$
\left(\frac{\partial}{\partial\eta}-\frac{1}{2N}\frac{\partial^2}{\partial\Lambda^2}\right)Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})=0. \qquad (52)
$$

Proof. The first equation is proven in the same way as the previous model (i.e. $\Lambda=0$). Here we focus on the proof of the second equation (52).

Consider the derivative with respect to Λ

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2}{\partial\Lambda^2}Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}})\\
&=\frac{1}{\mathrm{Vol}_N^3}\int_{\mathcal{H}_N^{\otimes 3}}dXdYdTN^2\sum_{i\geq 0}y^i\mathrm{e}^{-N\mathrm{Tr}W_{y,\eta,\alpha,\beta}(X,Y,T;\Lambda_1,\Lambda_2,\Lambda)}\\
&\qquad\qquad\times\Big\{\alpha_i^{(1)}X^iY^i+\alpha_i^{(2)}Y^iX^i+\beta_i^{(1)}(XY)^i+\beta_i^{(2)}(YX)^i\Big\}\\
&\qquad\qquad\times\Big((T-\eta^{-1/2}\Lambda)T+T(T-\eta^{-1/2}\Lambda)\Big)\Big]\Big\},
\end{aligned}
$$

where $T = M + \eta^{-1/2}\Lambda$ and

$$
\begin{aligned}
&W_{y,\eta,\alpha,\beta}(X,Y,T;\Lambda_1,\Lambda_2,\Lambda) \\
&= (X - y^{-1/2}\Lambda_1)(Y - y^{-1/2}\Lambda_2) + \frac{1}{2}(T - \eta^{-1/2}\Lambda)^2 \\
&\quad - \sum_{i\geq 0} y^i(\alpha_i^{(1)} T X^i Y^i T + \alpha_i^{(2)} T Y^i X^i T) \\
&\quad - \sum_{i\geq 0} y^i(\beta_i^{(1)} T (XY)^i T + \beta_i^{(2)} T (YX)^i T).
\end{aligned}
$$

The derivative with respect to η is given by

$$
\begin{aligned}
&\frac{\partial}{\partial\eta} Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r_{ij;p}}) \\
&= \frac{1}{\mathrm{Vol}_N^3} \int_{\mathcal{H}_N^{\otimes 3}} dA dB dM \ \frac{N}{2} \sum_{i\geq 0} y^i \mathrm{e}^{-N\mathrm{Tr}V_{y,\zeta,\alpha,\beta}(A,B;\Lambda_1,\Lambda_2,\Lambda)} \\
&\quad \times \mathrm{Tr}\Bigg[\Bigg\{ \alpha_i^{(1)}(A + y^{-1/2}\Lambda_1)^i (B + y^{-1/2}\Lambda_2)^i \\
&\qquad + \alpha_i^{(2)}(B + y^{-1/2}\Lambda_2)^i (A + y^{-1/2}\Lambda_1)^i \\
&\qquad + \beta_i^{(1)}\big((A + y^{-1/2}\Lambda_1)(B + y^{-1/2}\Lambda_2)\big)^i \\
&\qquad + \beta_i^{(2)}\big((B + y^{-1/2}\Lambda_2)(A + y^{-1/2}\Lambda_1)\big)^i \Bigg\} \\
&\qquad \times (M(M + \eta^{-1/2}\Lambda) + (M + \eta^{-1/2}\Lambda)M)\Bigg].
\end{aligned}
$$

Comparing these two results, we obtain the heat equation (51).

□

For the initial condition with $y = 0$ and $\eta = 0$, we find

$$
\begin{aligned}
&Z_N(y=0,\eta=0;\{\alpha_i^{(1)}\},\{\alpha_i^{(2)}\},\{\beta_i^{(1)}\},\{\beta_i^{(2)}\},\{r_{\{i\},\{j\};\{K\},p}\}) \\
&= \exp\left(\sum_{i\geq 0} \mathrm{Tr}\Lambda(\alpha_i^{(1)}(\Lambda_1^i\Lambda_2^i) + \alpha_i^{(2)}(\Lambda_2^i\Lambda_1^i) + \beta_i^{(1)}(\Lambda_1\Lambda_2)^i + \beta_i^{(2)}(\Lambda_2\Lambda_1)^i)\Lambda\right).
\end{aligned}
$$

The initial condition that keeps η can also be considered as follows:

$$
\begin{aligned}
&Z_N(y=0,\eta;\{\alpha_i^{(1)}\},\{\alpha_i^{(2)}\},\{\beta_i^{(1)}\},\{\beta_i^{(2)}\},\{r_{\{i\},\{j\};\{K\},p}\}) \\
&= \frac{1}{\mathrm{Vol}(\mathcal{H}_N)} \int_{\mathcal{H}_N} dM \ \exp\Bigg[-N\mathrm{Tr}\Bigg\{\frac{M^2}{2} - (M + \eta^{-1/2}\Lambda)\Big(\alpha_i^{(1)}(\Lambda_1^i\Lambda_2^i) + \alpha_i^{(2)}(\Lambda_2^i\Lambda_1^i) \\
&\qquad + \beta_i^{(1)}(\Lambda_1\Lambda_2)^i + \beta_i^{(2)}(\Lambda_2\Lambda_1)^i\Big)(M + \eta^{-1/2}\Lambda)\Bigg\}\Bigg].
\end{aligned}
$$

Finally, we express the heat equations as the cut-and-join equations. The indexing of r makes the notation cumbersome, but a systematic computation gives the following result.

Theorem 4.10. *Let L_0 and L_2 denote the derivatives following differential operators;*

$$L_0 = \sum_{p\geq 1}\sum_{\{K\}}\sum_{\{i\},\{j\}}\sum_{q=1}^{p}\sum_{r=1}^{p}\sum_{L=1}^{K_q}\sum_{M=1}^{K_r}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{m=0}^{j_M^{(r)}-1}$$
$$r_{(i_1^{(r)},\dots,i_M^{(r)},\ell,i_{L+1}^{(q)},\dots,i_{K_q}^{(q)}:i_1^{(q+1)},\dots:\cdots:\dots,i_{K_{r-1}}^{(r-1)})}$$
$$,(j_1^{(r)},\dots,j_{M-1}^{(r)},j_M^{(r)}-m-1,j_L^{(q)},\dots,j_{K_q}^{(q)}:j_1^{(q+1)},\dots:\cdots:\dots,j_{K_{r-1}}^{(r-1)})$$
$$\times r_{(i_1^{(q)},\dots,i_{L-1}^{(q)},i_L^{(q)}-\ell-1,i_{M+1}^{(r)},\dots,i_{K_r}^{(r)}:i_1^{(r+1)},\dots:\cdots:\dots,i_{K_{q-1}}^{(q-1)})}$$
$$,(j_1^{(q)},\dots,j_{L-1}^{(q)},m,j_{M+1}^{(r)},\dots,j_{K_r}^{(r)}:j_1^{(r+1)},\dots,:\cdots:\dots,j_{K_{q-1}}^{(q-1)})$$
$$\times\frac{\partial}{\partial r_{(i_1^{(1)},\dots:\cdots:\dots,i_{K_p}^{(p)}),(j_1^{(1)},\dots:\cdots:\dots,j_{K_p}^{(p)})}},$$

$$L_2 = \sum_{p,u\geq 1}\sum_{\{K\},\{V\}}\sum_{\{i,j\}}\sum_{\{s,t\}}\sum_{q=1}^{p}\sum_{w=1}^{u}\sum_{L=1}^{K_q}\sum_{R=1}^{V_w}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{b=0}^{t_R^{(w)}-1}$$
$$r_{(s_1^{(w)},\dots,s_R^{(w)},\ell,i_{L+1}^{(q)},\dots,i_{K_q}^{(q)}:i_1^{(q+1)},\dots:\cdots:\dots,i_{K_{q-1}}^{(q-1)}:}$$
$$i_1^{(q)},\dots,i_{L-1}^{(q)},i_L^{(q)}-\ell-1,s_{R+1}^{(w)},\dots,s_{V_w}^{(w)}:s_1^{(w+1)},\dots:\cdots:\dots,s_{V_{w-1}}^{(w-1)}:)$$
$$,(t_1^{(w)},\dots,t_{R-1}^{(w)},t_R^{(w)}-b-1,j_L^{(q)},\dots,j_{K_q}^{(q)}:j_1^{(q+1)},\dots:\cdots:\dots,j_{K_{q-1}}^{(q-1)}:$$
$$j_1^{(q)},\dots,j_{L-1}^{(q)},b,t_{R+1}^{(w)},\dots,t_{V_w}^{(w)}:t_1^{(w+1)},\dots:\cdots:\dots,t_{V_{w-1}}^{(w-1)}:)$$
$$\times\frac{\partial^2}{\partial r_{(i_1^{(1)},\dots:\cdots:\dots,i_{K_p}^{(p)}),(j_1^{(1)},\dots:\cdots:\dots,j_{K_p}^{(p)})}\partial r_{(s_1^{(1)},\dots:\cdots:\dots,s_{V_u}^{(u)}),(t_1^{(1)},\dots:\cdots:\dots,t_{V_u}^{(u)})}}.$$

Let M_0 and M_2 denote the following differential operators;

$$M_0 = \frac{1}{2}\sum_{p\geq 1}\sum_{\{K\}}\sum_{\{i\},\{j\}}\sum_{q=1}^{p}\sum_{r=1}^{p}$$
$$r_{(i_1^{(r-1)},\dots,i_{K_{r-1}}^{(r-1)},i_1^{(q)},\dots,i_{K_q}^{(q)}:i_1^{(q+1)},\dots,:\cdots:,\dots,i_{K_{r-2}}^{(r-2)}:)}$$
$$,(j_1^{(r-1)},\dots,j_{K_{r-1}}^{(r-1)},j_1^{(q)},\dots,j_{K_q}^{(q)}:j_1^{(q+1)},\dots,:\cdots:,\dots,j_{K_{r-2}}^{(r-2)}:)$$
$$\times r_{(i_1^{(q-1)},\dots,i_{K_{q-1}}^{(q-1)},i_1^{(r)},\dots,i_{K_r}^{(r)}:i_1^{(r+1)},\dots,:\cdots:,\dots,i_{K_{q-2}}^{(q-2)}:)}$$
$$,(j_1^{(q-1)},\dots,j_{K_{q-1}}^{(q-1)},j_1^{(r)},\dots,j_{K_r}^{(r)}:j_1^{(r+1)},\dots,:\cdots:,\dots,j_{K_{q-2}}^{(q-2)}:)$$
$$\times\frac{\partial}{\partial r_{(i_1^{(1)},\dots:\cdots:\dots,i_{K_p}^{(p)}),(j_1^{(1)},\dots:\cdots:\dots,j_{K_p}^{(p)})}},$$

$$M_2 = \frac{1}{2}\sum_{p,u\geq 1}\sum_{\{K\},\{V\}}\sum_{\{i\},\{j\}}\sum_{\{s\},\{t\}}\sum_{q=1}^{p}\sum_{w=1}^{u}$$
$$r_{(s_1^{(w-1)},\dots,s_{V_{w-1}}^{(w-1)},i_1^{(q)},\dots,i_{K_q}^{(q)}:i_1^{(q+1)},\dots:\cdots:,\dots,i_{K_{q-2}}^{(q-2)}:}$$
$$i_1^{(q-1)},\dots,i_{K_{q-1}}^{(q-1)},s_1^{(w)},\dots,s_{V_w}^{(w)}:s_1^{(w+1)},\dots:\cdots:\dots,s_{V_{w-2}}^{(w-2)}:)$$
$$,(t_1^{(w-1)},\dots,t_{V_{w-1}}^{(w-1)},j_1^{(q)},\dots,j_{K_q}^{(q)}:j_1^{(q+1)},\dots:\cdots:,\dots,j_{K_{q-2}}^{(q-2)}:$$
$$j_1^{(q-1)},\dots,j_{K_{q-1}}^{(q-1)},t_1^{(w)},\dots,t_{V_w}^{(w)}:t_1^{(w+1)},\dots:\cdots:\dots,t_{V_{w-2}}^{(w-2)}:)$$
$$\times\frac{\partial^2}{\partial r_{(i_1^{(1)},\dots:\cdots:\dots,i_{K_p}^{(p)}),(j_1^{(1)},\dots:\cdots:\dots,j_{K_p}^{(p)})}\partial r_{(s_1^{(1)},\dots:\cdots:\dots,s_{V_u}^{(u)}),(t_1^{(1)},\dots:\cdots:\dots,t_{V_u}^{(u)})}}.$$

The heat equations (51) and (52) can be rewritten as the cut-and-join equa-

tions:

$$\frac{\partial Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r}_{\boldsymbol{ij};p})}{\partial y}$$
$$= \left(L_0 + \frac{1}{N^2}L_2\right) Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r}_{\boldsymbol{ij};p}), \tag{53}$$
$$\frac{\partial Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r}_{\boldsymbol{ij};p})}{\partial \eta}$$
$$= \left(M_0 + \frac{1}{N^2}M_2\right) Z_N(y,\eta;\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)},\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\boldsymbol{r}_{\boldsymbol{ij};p}). \tag{54}$$

Proof. First we will derive L_0 and L_2 operators from the chain rule. The L_0 operator comes from the following derivative:

$$\begin{aligned}
&\mathrm{Tr}\frac{\partial^2 r_{\boldsymbol{ij};p}}{\partial\Lambda_1\partial\Lambda_2} \\
&= \frac{1}{N}\sum_{q=1}^{p}\sum_{r=1}^{p}\sum_{L=1}^{K_q}\sum_{M=1}^{K_r}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{m=0}^{j_M^{(r)}-1} \\
&\quad \mathrm{Tr}\left(\Lambda_1^{\ell}\Lambda_2^{j_L^{(q)}}\Lambda_1^{i_{L+1}^{(q)}}\Lambda_2^{j_{L+1}^{(q)}}\cdots\Lambda_1^{i_{K_q}^{(q)}}\Lambda_2^{j_{K_q}^{(q)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(r)}}\Lambda_2^{j_1^{(r)}}\cdots\Lambda_1^{i_M^{(r)}}\Lambda_2^{j_M^{(r)}-m-1}\right) \\
&\quad \times\mathrm{Tr}\left(\Lambda_2^{m}\Lambda_1^{i_{M+1}^{(r)}}\Lambda_2^{j_{M+1}^{(r)}}\cdots\Lambda_1^{i_{K_r}^{(r)}}\Lambda_2^{j_{K_r}^{(r)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(q)}}\Lambda_2^{j_1^{(q)}}\cdots\Lambda_1^{i_L^{(q)}-\ell-1}\right) \\
&= N\sum_{q=1}^{p}\sum_{r=1}^{p}\sum_{L=1}^{K_q}\sum_{M=1}^{K_r}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{m=0}^{j_M^{(r)}-1} \\
&\quad r_{(i_1^{(r)},\ldots,i_M^{(r)},\ell,i_{L+1}^{(q)},\ldots,i_{K_q}^{(q)}:i_1^{(q+1)},\ldots:\cdots:\ldots,i_{K_{r-1}}^{(r-1)})} \\
&\quad\quad {}_{,(j_1^{(r)},\ldots,j_{M-1}^{(r)},j_M^{(r)}-m-1,j_L^{(q)},\ldots,j_{K_q}^{(q)}:j_1^{(q+1)},\ldots:\cdots:\ldots,j_{K_{r-1}}^{(r-1)})} \\
&\quad \times r_{(i_1^{(q)},\ldots,i_{L-1}^{(q)},i_L^{(q)}-\ell-1,i_{M+1}^{(r)},\ldots,i_{K_r}^{(r)}:i_1^{(r+1)},\ldots:\cdots:\ldots,i_{K_{q-1}}^{(q-1)})} \\
&\quad\quad {}_{,(j_1^{(q)},\ldots,j_{L-1}^{(q)},m,j_{M+1}^{(r)},\ldots,j_{K_r}^{(r)}:j_1^{(r+1)},\ldots,:\cdots:\ldots,j_{K_{q-1}}^{(q-1)})} .
\end{aligned}$$

The L_2 operator is from

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial r_{\boldsymbol{ij};p}}{\partial \Lambda_1}\frac{\partial r_{\boldsymbol{st};u}}{\partial \Lambda_2}\\
&=\frac{1}{N^2}\sum_{q=1}^{p}\sum_{w=1}^{u}\sum_{L=1}^{K_q}\sum_{R=1}^{V_w}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{b=0}^{t_R^{(w)}-1}\\
&\quad\mathrm{Tr}\Big(\Lambda_1^{\ell}\Lambda_2^{j_L^{(q)}}\Lambda_1^{i_{L+1}^{(q)}}\Lambda_2^{j_{L+1}^{(q)}}\cdots\Lambda_1^{i_{K_q}^{(q)}}\Lambda_2^{j_{K_q}^{(q)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(q)}}\Lambda_2^{j_1^{(q)}}\cdots\Lambda_1^{i_L^{(q)}-\ell-1}\\
&\qquad\cdot\Lambda_2^{b}\Lambda_1^{s_{R+1}^{(w)}}\Lambda_2^{t_{R+1}^{(w)}}\cdots\Lambda_1^{s_{V_w}^{(w)}}\Lambda_2^{t_{V_w}^{(w)}}\Lambda\cdots\Lambda\Lambda_1^{s_1^{(w)}}\Lambda_2^{t_1^{(w)}}\cdots\Lambda_1^{s_R^{(w)}}\Lambda_2^{t_R^{(w)}-b-1}\Big)\\
&=\frac{1}{N}\sum_{q=1}^{p}\sum_{w=1}^{u}\sum_{L=1}^{K_q}\sum_{R=1}^{V_w}\sum_{\ell=0}^{i_L^{(q)}-1}\sum_{b=0}^{t_R^{(w)}-1}\\
&\quad r_{\substack{(s_1^{(w)},\ldots,s_R^{(w)},\ell,i_{L+1}^{(q)},\ldots,i_{Kq}^{(q)}:i_1^{(q+1)},\ldots:\cdots:\ldots,i_{K_{q-1}}^{(q-1)}:\\ i_1^{(q)},\ldots,i_{L-1}^{(q)},i_L^{(q)}-\ell-1,s_{R+1}^{(w)},\ldots,s_{Vw}^{(w)}:s_1^{(w+1)},\ldots:\cdots:\ldots,s_{V_{w-1}}^{(w-1)}:)\\ ,(t_1^{(w)},\ldots,t_{R-1}^{(w)},t_R^{(w)}-b-1,j_L^{(q)},\ldots,j_{Kq}^{(q)}:j_1^{(q+1)},\ldots:\cdots:\ldots,j_{K_{q-1}}^{(q-1)}:\\ j_1^{(q)},\ldots,j_{L-1}^{(q)},b,t_{R+1}^{(w)},\ldots,t_{Vw}^{(w)}:t_1^{(w+1)},\ldots,:\cdots:\ldots,t_{V_{w-1}}^{(w-1)}:)}}.
\end{aligned}
$$

For the second heat equation, the M_0 operator is from

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial^2 r_{\boldsymbol{ij};p}}{\partial \Lambda^2}\\
&=\frac{1}{N}\sum_{q=1}^{p}\sum_{r=1}^{p}\mathrm{Tr}\Big(\Lambda_1^{i_1^{(q)}}\Lambda_2^{j_1^{(q)}}\cdots\Lambda_1^{i_{K_q}^{(q)}}\Lambda_2^{j_{K_q}^{(q)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(r-1)}}\Lambda_2^{j_1^{(r-1)}}\cdots\Lambda_1^{i_{K_{r-1}}^{(r-1)}}\Lambda_2^{j_{K_{r-1}}^{(r-1)}}\Big)\\
&\qquad\times\mathrm{Tr}\Big(\Lambda_1^{i_1^{(r)}}\Lambda_2^{j_1^{(r)}}\cdots\Lambda_1^{i_{K_r}^{(r)}}\Lambda_2^{j_{K_r}^{(r)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(q-1)}}\Lambda_2^{j_1^{(q-1)}}\cdots\Lambda_1^{i_{K_{q-1}}^{(q-1)}}\Lambda_2^{j_{K_{q-1}}^{(q-1)}}\Big)\\
&=N\sum_{q=1}^{p}\sum_{r=1}^{p}r_{\substack{(i_1^{(r-1)},\ldots,i_{K_{r-1}}^{(r-1)},i_1^{(q)},\ldots,i_{Kq}^{(q)}:i_1^{(q+1)},\ldots,:\cdots:,\ldots,i_{K_{r-2}}^{(r-2)}:)\\ ,(j_1^{(r-1)},\ldots,j_{K_{r-1}}^{(r-1)},j_1^{(q)},\ldots,j_{Kq}^{(q)}:j_1^{(q+1)},\ldots,:\cdots:,\ldots,j_{K_{r-2}}^{(r-2)}:)}}\\
&\qquad\times r_{\substack{(i_1^{(q-1)},\ldots,i_{K_{q-1}}^{(q-1)},i_1^{(r)},\ldots,i_{K_r}^{(r)}:i_1^{(r+1)},\ldots,:\cdots:,\ldots,i_{K_{q-2}}^{(q-2)}:)\\ ,(j_1^{(q-1)},\ldots,j_{K_{q-1}}^{(q-1)},j_1^{(r)},\ldots,j_{K_r}^{(r)}:j_1^{(r+1)},\ldots,:\cdots:,\ldots,j_{K_{q-2}}^{(q-2)}:)}}.
\end{aligned}
$$

Finally, the M_2 operator is from

$$
\begin{aligned}
&\mathrm{Tr}\frac{\partial r_{\boldsymbol{ij};p}}{\partial \Lambda}\frac{\partial r_{\boldsymbol{st};u}}{\partial \Lambda}\\
&=\frac{1}{N^2}\sum_{q=1}^{p}\sum_{w=1}^{u}\mathrm{Tr}\Big(\Lambda_1^{i_1^{(q)}}\Lambda_2^{j_1^{(q)}}\cdots\Lambda_1^{i_{K_q}^{(q)}}\Lambda_2^{j_{K_q}^{(q)}}\Lambda\cdots\Lambda\Lambda_1^{i_1^{(q-1)}}\Lambda_2^{j_1^{(q-1)}}\cdots\Lambda_1^{i_{K_{q-1}}^{(q-1)}}\Lambda_2^{j_{K_{q-1}}^{(q-1)}}\\
&\qquad\times\Lambda_1^{s_1^{(w)}}\Lambda_2^{t_1^{(w)}}\cdots\Lambda_1^{s_{V_w}^{(w)}}\Lambda_2^{t_{V_w}^{(w)}}\Lambda\cdots\Lambda\Lambda_1^{s_1^{(w-1)}}\Lambda_2^{t_1^{(w-1)}}\cdots\Lambda_1^{s_{V_{w-1}}^{(w-1)}}\Lambda_2^{t_{V_{w-1}}^{(w-1)}}\Big)\\
&=\frac{1}{N}\sum_{q=1}^{p}\sum_{w=1}^{u}r_{\substack{(s_1^{(w-1)},\ldots,s_{V_{w-1}}^{(w-1)},i_1^{(q)},\ldots,i_{Kq}^{(q)}:i_1^{(q+1)},\ldots:\cdots:,\ldots,i_{K_{q-2}}^{(q-2)}:\\ i_1^{(q-1)},\ldots,i_{K_{q-1}}^{(q-1)},s_1^{(w)},\ldots,s_{Vw}^{(w)}:s_1^{(w+1)},\ldots:\cdots:\ldots,s_{V_{w-2}}^{(w-2)}:)\\ ,(t_1^{(w-1)},\ldots,t_{V_{w-1}}^{(w-1)},j_1^{(q)},\ldots,j_{Kq}^{(q)}:j_1^{(q+1)},\ldots:\cdots:,\ldots,j_{K_{q-2}}^{(q-2)}:\\ j_1^{(q-1)},\ldots,j_{K_{q-1}}^{(q-1)},t_1^{(w)},\ldots,t_{Vw}^{(w)}:t_1^{(w+1)},\ldots:\cdots:\ldots,t_{V_{w-2}}^{(w-2)}:)}}.
\end{aligned}
$$

□

We note the first cut-and-join equation (53) expresses the cut/join manipulation of the hydrogen bonds, while the second cut-and-join equation (54) is for loops (or turns) in the backbones.

5 Topology of protein β-sheets

5.1 β-sheet topology

The α-helix and the β-sheet are two common protein secondary structures. While the α-helix is essentially a local structure with the participating residues all lying together along the backbone, the β-sheet involves interactions between residues which are far apart in the backbone (section 4.1). It is also more heterogeneous as a structure, consisting of both parallel and anti-parallel configurations of the participating β-strands. Furthermore, β-sheet has an intrinsic structural flexibility compared to α-helix, complicating the structural analyses [30]. A better understanding of their structures and foldings is therefore crucial, if we are to understand the folding mechanism of entire proteins.

The configurations of β-strands in a protein, often called β-sheet topologies, have been studied since the 1970's [79]. Early studies ([79, 78, 87]) have identified some general rules (such as the preference for the right-handedness in parallel β-sheets) from investigation of individual proteins. As the amount of available data increased, studies have used computer programmes to survey the database and found frequent patterns in the β-strand configurations [99, 80]. The information can be used to filter and rank a series of candidate structures by computing probabilities for different patterns [80]. Another approach is to assign pseudoenergy to each pair of β-strand residues and solve the β-sheet topology prediction problem as an optimisation problem [29]. At least one study [44] has compared the two methods, and found that the latter's performance to be better. One may also combine the two methods by, for example, forbidding certain β-strand configurations that are not found in the database [89], or by incorporating the two in Bayesian modelling [18]. Other studies used integer programming techniques to predict β-sheet topologies [81, 34].

In order to study β-sheet topology of proteins, we introduce a new model inspired by the protein fatgraph model described in section 4, which we call protein metastructure. This model greatly simplifies the study of β-sheet topologies by amalgamating consecutive residues belonging to the same secondary structure, but still retains the information needed to understand the configuration of β-strands. We give a detailed definition in section 5.2. Furthermore, each metastructure corresponds to a fatgraph, and this transition to fatgraphs allows us to compute topological invariants such as the number of boundary components and the genus associated to each protein. The details of this correspondence are described in section 2.1. Compared to the model described in section 4, our construction is much simpler, and only takes into account the hydrogen bonds that are part of β-sheets. In the following sections, we will analyse the topology of fatgraphs associated to proteins and suggest potential applications in the study of β-sheet topology.

5.2 Protein metastructure

Given a protein, its primary structure is the sequence of amino acids in the polypeptide chain. There are 20 different amino acids in the standard gene code, so a primary structure can be expressed as a finite word in an alphabet with 20 letters;

EKKSINECDLKGKKVLIRVDFNVPVKNGKITNDYRIRSALPTLKKV...

The secondary structure of a protein can be defined as a set of local substructures, most frequent of which are α-helices and β-sheets. The DSSP-algorithm [51] is an algorithm commonly used to classify residues into 3 or 7 secondary structure classes. When used (with 3-class output) on the above protein it produces a word in an alphabet with 3 letters;

$$\gamma\gamma\gamma\gamma\alpha\alpha\alpha\gamma\gamma\gamma\gamma\gamma\gamma\beta\beta\beta\beta\beta\beta\gamma\gamma\gamma\gamma\gamma\beta\beta\gamma\gamma\beta\beta\gamma\gamma\gamma\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha... \quad (55)$$

Here we used the letter α for [H]elices, β for [S]heets, and γ for [C]oils. With this reduction from 20 to 3 classes, we begin to see some patterns in the proportion of these classes in proteins. There are, for example, few proteins which contain less than 25% γ residues, or more than 75% of any one class (figure 35a). This can be explained by the rigidness of helix and sheet structures; a protein composed (almost) exclusively of α or β residues will not have the necessary flexibility to bend and fold into its native structure. For that to occur, a certain proportion of γ residues are required. On the other hand, too much γ residues would most likely result in lack of stability and will be energetically unfavourable. The largest concentration appears to be around 30$\sim$50% α, 10$\sim$30% β, and 30$\sim$50% γ residues (figure 35a).

We now introduce *reduced secondary structure sequence* by reducing each segment of identical letters in a secondary structure sequence (55) to a single letter;

$$\gamma\alpha\gamma\beta\gamma\beta\gamma\beta\gamma\alpha...$$

Not surprisingly, the distribution of proportions of the 3 classes in such reduced sequences are concentrated around $\gamma = 50\%$ (figure 35b), since the reduced sequences are mostly sequences of $\gamma\alpha$ and $\gamma\beta$ by construction.

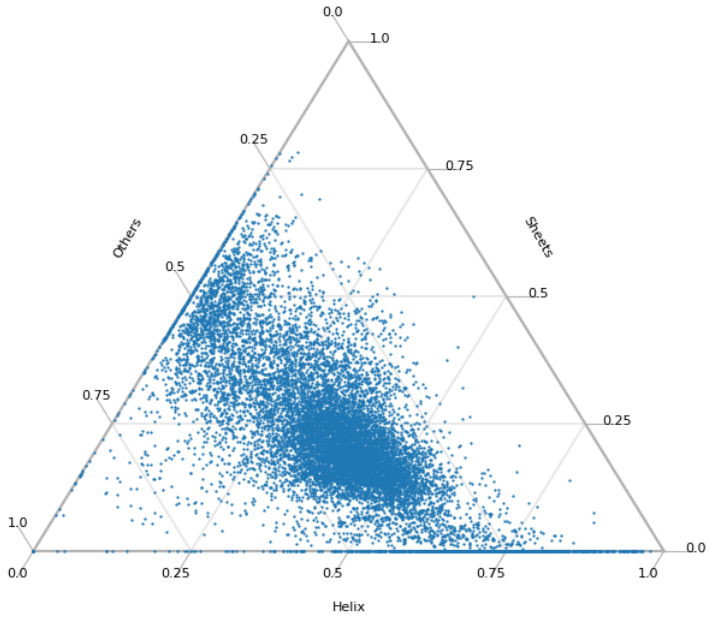
In a reduced sequence, each letter β corresponds to a β-strand. We may therefore add an additional data to a reduced sequence to specify β-sheet structure of the protein. To do this, we define *protein metastructure* as the triple (r, P, A), where r is a finite word in an alphabet of three letters, α, β and γ, and P and A are sets of pairs of integers (i, j) for some $1 \leq i < j \leq s$, where s is the number of letter β in r. We also put a further condition, that $P \cap A = \emptyset$. Then for a given protein, we obtain its metastructure by setting r to be the reduced sequence, and populating P and A as follows;

1. Number the letters β in r along the backbone, starting from the N-terminus.

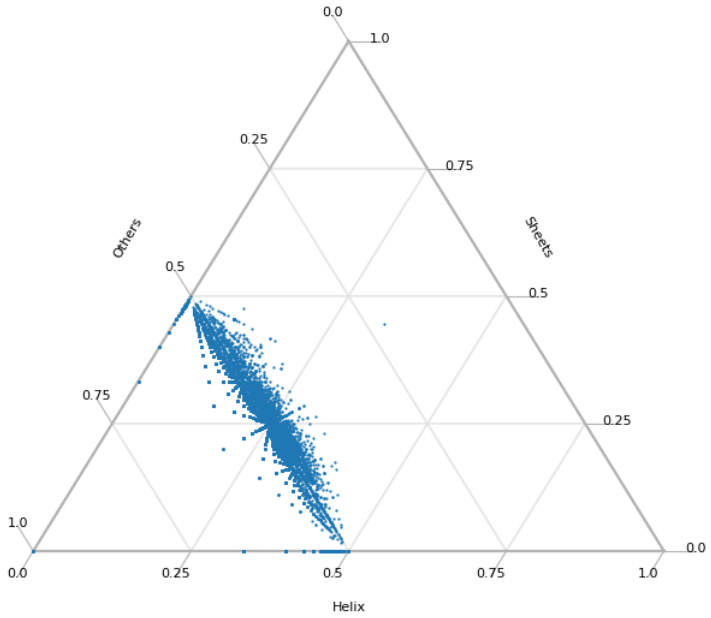
2. Identify all pairs (i, j), where there is at least one hydrogen bond between ith and jth strands.

3. Let I be the set of all pairs (i, j) identified in the previous step. Partition I into two sets P and A, where P consists of all parallel pairs and A all anti-parallel pairs.

If there is only a single bond between two strands, thus making it impossible to determine the configuration between the two, we extend the strands by up to three residues. If it is still not possible to determine the configuration (because the extended strands has a single bond between them), then we assign the pair to P, as parallel configuration. This is because the standard anti-parallel



(a) Proportion of 3 classes in secondary structure sequences



(b) Proportion of 3 classes in reduced sequences

Figure 35: Proportion of 3 classes in 13107 selected proteins

configuration requires two hydrogen bonds between a pair of residues, thus making it less likely that there is only one hydrogen bond present.

Let $\mathcal{S}$ be the set of all possible metastructures, and let $\mathcal{S}_{\text{bif}} \subset \mathcal{S}$ be the subset consisting of all metastructures, where at least one β-strand is connected to more than 2 other strands (bifurcations). Similarly, let $\mathcal{S}_{\text{bar}}$ be the subset of metastructures with β-barrels, and $\mathcal{S}_{\text{iso}}$ be the subset with at least one unpaired β-strand. Consider $\tilde{\mathcal{S}} = \mathcal{S} \setminus (\mathcal{S}_{\text{bif}} \cup \mathcal{S}_{\text{bar}} \cup \mathcal{S}_{\text{iso}})$. For each $s \in \tilde{\mathcal{S}}$, we can associate a metastructure motif diagram (figure 36) as follows;

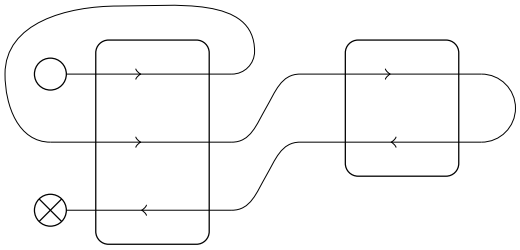


Figure 36: An example metastructure motif diagram. The associated metastructure may be $(\gamma\beta\gamma\alpha\gamma\beta\gamma\beta\gamma\beta\gamma\beta\gamma, \{(1,2)\}, \{(3,4),(2,5)\})$

1. Each β-strand is denoted by a straight line segment with an arrowhead in the middle.
2. If $(i,j) \in P$, draw ith and jth strands next to each other, with arrowheads on both segments pointing the same direction.
3. If $(i',j') \in A$, draw $i'$the and j'th strands next to each other, with arrowheads pointing the opposite direction.
4. Draw a "sheet" around each stack of strands.
5. Connect the strands, from the 1st to last, following the directions of arrowheads, and avoiding the interior of the sheets.
6. N-terminus is denoted by $\circ$, and C-terminus is denoted by $\otimes$.
7. Note for each sheet, we have a choice of 2 strands to draw on the top, (see figure 37; orientation of all other strands are then decided by parallel/anti-parallel configurations). We can make this canonical by requiring that the top strand comes before the bottom strand in the backbone-ordering.

We note that the metastructure of a protein with n β-strands can be recorded in an $n \times n$ matrix, whose entries are either 0 or 1. This can be done by setting (i,j)'th entry to 1 if the i'th and j'th strands are paired in the parallel configuration, and setting (j,i)'th entry to 1 if the pairing is anti-parallel. All other entries (where there is no pairing observed) are set to 0. We call this matrix $\mathbb{P}$ the protein's pairing matrix (figure 38). The 1's in the upper-triangular part show parallel pairings, and the 1's in the lower-triangular part show anti-parallel pairings. The number of paired strands the ith strand has can be computed as the total number of 1 cells in the ith row and column. A β-sheet manifests itself

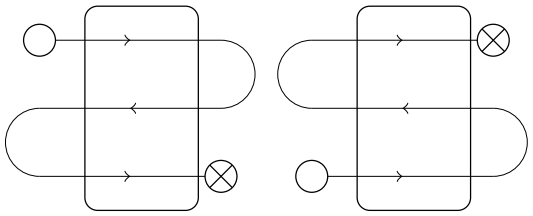


Figure 37: Two metastructure diagrams with two choices of the top-strand.

	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	1	0	0
5	0	1	0	0	0

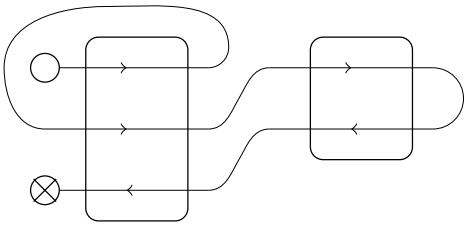


Figure 38: Pairing matrix and the corresponding protein metastructure diagram. The 1 in (1,2)-th entry corresponds to the parallel configuration between the first and the second strand in the backbone, and the two 1's in the lower-triangular part correspond to the anti-parallel configurations between the third and the fourth strands, and the second and the fifth strands.

as a "chain" of strands, with the edge strand having only one non-zero entry in the corresponding row or column.

If we have more detailed information about the protein structure, it is possible to make the choice of the top strand for each sheet more natural;

1. Choose a β-sheet and identify the first strand (ordered along the backbone) in the sheet.

2. On the first strand, identify the first hydrogen bond (ordered along the backbone) that participate in a β-strand pairing.

3. If the first hydrogen bond is connected to a carboxyl oxygen acceptor, the strand paired to the first strand is drawn to the left of the first strand. If the hydrogen bond is connected to an amino hydrogen donor, it is drawn to the right of the first strand.

4. Configuration of the remaining strands in the sheet is decided by the information on parallel/anti-parallel pairings.

5. Repeat for all β-sheets in the protein.

We call the protein metastructure with this extra information the *extended metastructure*. An extended metastructure can again be defined as the triple (r, P, A), with the reduced sequence as r, but with P and A now being sets of (unordered) pairs (i, j) for some $-s \leq i, j \leq s$, where s is the number of strands.

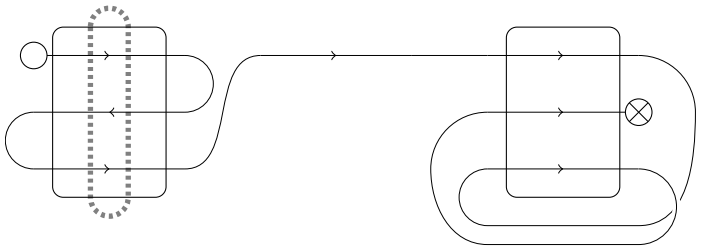


Figure 39: An extended metastructure diagram. A barrel is denoted by the dashed circle around the first sheet. Note the isolated strand in the middle.

	1	2	3	4	5	6	7
1	0	0	2	0	0	0	0
2	3	0	0	0	0	0	0
3	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	4
6	0	0	0	0	0	0	2
7	0	0	0	0	0	0	0

Figure 40: A pairing matrix corresponding to the extended metastructure diagram in figure 39. The first sheet is a barrel with strands 1, 2, and 3. The strands 1 and 2 are paired in anti-parallel configuration, and the bond between strands 1 and 2 is to the right of 1 and 2. The corresponding entry is the number 3 in (2,1).

If $i < 0$, we connect the left side of the ith strand, and if $i > 0$, we connect the right side of the ith strand. We will also allow barrels and isolated strands in the extended metastructures (figure 39). The pairing matrix $\mathbb{P}$ for an extended metastructure has entries in integers $\{0, 1, 2, 3, 4\}$. For a pair of strands i, j, the pairing information is recorded as follows;

$$\mathbb{P}_{i,j} = \begin{cases} 0 & \text{if no bond between } i\text{th and } j\text{th strands} \\ 1 & \text{if bond between the left side of } i \text{ and the left side of } j \\ 2 & \text{if bond between the left side of } i \text{ and the right side of } j \\ 3 & \text{if bond between the right side of } i \text{ and the right side of } j \\ 4 & \text{if bond between the right side of } i \text{ and the left side of } j \end{cases}$$

all in the parallel configuration if $i < j$ and in the anti-parallel configuration if $j < i$ (see figure 40). Note the definition allows a single strand forming a barrel, which does not occur in nature. In a pairing matrix for an extended metastructure, an isolated strand can be seen as zero row and column; the ith

strand is isolated (has no pairing), if and only if the ith row and the ith column only contain 0. A β-barrel is a circular "chain" of strands without an edge strand (see figure 40).

Note, if $\mathcal{T}$ is the set of metastructure diagrams, the map $\varphi : \tilde{\mathcal{S}} \to \mathcal{T}$ described above corresponds to "forgetting" r in $(r, P, A) \in \tilde{\mathcal{S}}$.

5.3 Protein metastructure and fatgraph

In order to understand topological characteristics of protein metastructures, we need to pass from metastructure diagrams to topological surfaces. The main idea is to "thicken" the non-β segments in a given metastructure diagram to (untwisted) bands or ribbons, as in figure 41, to produce a fatgraph $\mathbb{D}$.

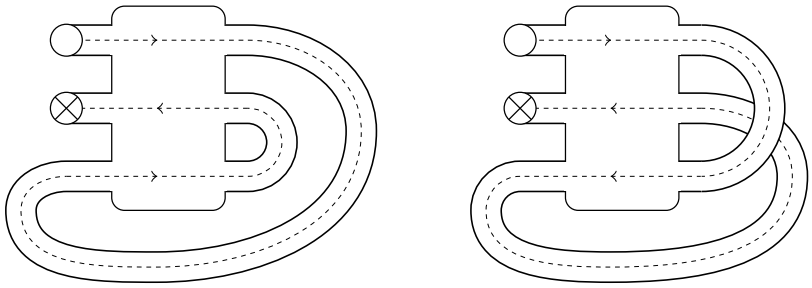


Figure 41: Thickening edges of metastructure diagrams to obtain fatgraphs (or more precisely, surfaces associated to fatgraphs). The surface on the left has genus 0, whereas the one on the right has genus 1.

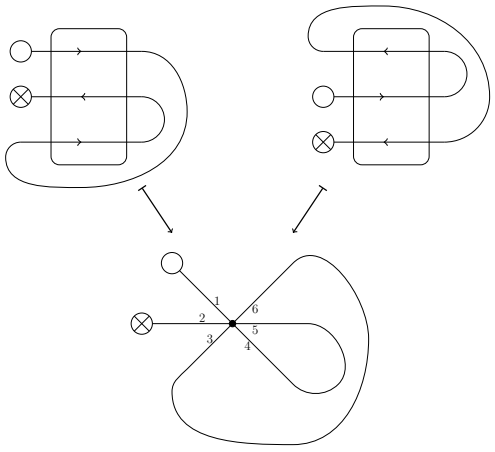


Figure 42: Construction of fatgraph from metastructure diagrams. Note the two different motifs result in an identical fatgraph.

We recall that a fatgraph $\mathbb{D}$ is a graph D together with a cyclic ordering of the incident half-edges at each vertex (section 2.1). It can be obtained from a metastructure diagram by contracting each sheet to a point, and ordering the

resulting half-edges at each vertex anti-clockwise from the N-terminus, or the starting end of the first strand in the sheet (figure 42). A fatgraph $\mathbb{D}$ gives rise to a unique (orientable) surface $X_{\mathbb{D}}$ by thickening each edge to a band and each vertex to a disc. As an orientable surface, it obeys the Euler's formula

$$\chi(X_{\mathbb{D}}) = v - e + n = 2 - 2g,$$

where v is the number of vertices (which correspond to the β-sheets in the metastructure diagram), e the number of edges or bands (corresponding to the non-β segments, excluding the N- and C-terminal segments), n the number of boundary components, and g the genus of $X_{\mathbb{D}}$.

Note this map ψ from $\mathcal{T}$ to the set Σ of fatgraphs with two marked half-edges is not injective (figure 42). Nonetheless the composition $\psi \circ \varphi$ allows us to compute topological invariants, such as genus and number of boundary components for protein metastructures.

5.4 Recursion relation for extended metastructure

We present a recursion relation for the extended protein metastructures. Let us consider a protein metastructure diagram as in figure 43.

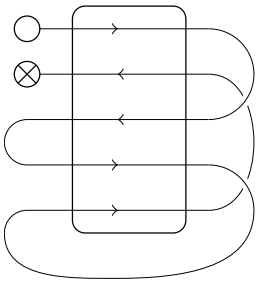


Figure 43: An example protein metastructure diagram

If we straighten the backbone and represent the strands as points on the backbone, we obtain a diagram shown in figure 44, where the pairings between strands are represented by arcs above and below the backbone. Each strand has left and right side, induced by the natural orientation of the backbone from the N-terminus to the C-terminus. We note that an anti-parallel pairing connects the same side of the two paired strands, while a parallel pairing connects different sides. Indeed, we see in figure 44, that it has two arcs that cross the backbone, and two that do not, representing parallel and anti-parallel pairings, respectively.

Furthermore, we may rotate the part of the diagram below the backbone by 180 degrees about the C-terminus (represented by the circle with a cross) to obtain a diagram where all the arcs representing the strand pairings are shown above the backbone (figure 45). Here the vertices to the left of the C-terminus (the centre of the backbone) represent the left side of the strands (ordered from the first to the fifth strand), while the vertices to the right represent the right side of the strands (ordered from the fifth to the first strand).

We note that neither the straightening of the backbone or the rotation of the lower half of the diagram change the genus or the number of boundary

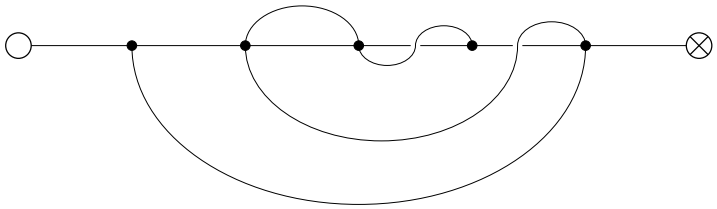


Figure 44: A metastructure diagram with the backbone as the horizontal, straight line.

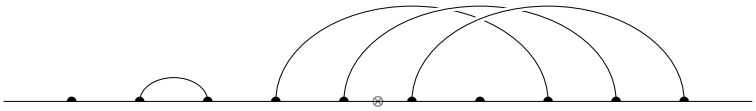


Figure 45: A metastructure diagram with all arcs above the straight backbone. The C-terminus is shown for reference.

components of the diagram. Therefore, the situation becomes exactly the same as for the RNA chord diagrams, as discussed in, for example, [3] or [15].

Let g, m, and p denote the genus, the number of strands and the number of strand pairings for a metastructure diagram. We also define the boundary point spectrum $\boldsymbol{n} = (n_0, n_1, n_2, \dots)$, where n_i is the number of boundary components that contain i unpaired sides. In the representation in figure 45, the number of unpaired sides is counted when a boundary component traverses a strand above the backbone. By considering the representation in figure 45, we find that the recursion relation has the same form as the one-backbone version of (34) [3];

$$
\begin{aligned}
&p\mathcal{N}_{g,m,p}(\boldsymbol{n}) = \\
&\quad \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=0}^{i}(i+2)(n_{i+2}+1)\mathcal{N}_{g,m,p-1}(\boldsymbol{n}-\boldsymbol{e}_j-\boldsymbol{e}_{i-j}+\boldsymbol{e}_{i+2}) + \\
&\quad \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=1}^{i+1}j(i+2-j)(n_j+1+\delta_{j,i+2-j}-\delta_{i,j})(n_{i+2-j}+1-\delta_{j,2})\times \\
&\qquad \mathcal{N}_{g-1,m,p-1}(\boldsymbol{n}+\boldsymbol{e}_j+\boldsymbol{e}_{i+2-j}-\boldsymbol{e}_i).
\end{aligned}
$$

Another, simpler recursion can be obtained by considering the enumeration of complete chord diagrams by Harer and Zagier (theorem 3.1;[45]). As before, let g, and p denote the genus and the number of strand pairings. The recursion relation for the complete chord diagrams is given by;

$$(p+1)\mathcal{C}_g(p) = 2(2p-1)\mathcal{C}_g(p-1) + (2p-1)(p-1)(2p-3)\mathcal{C}_{g-1}(p-2), \quad (56)$$

with $\mathcal{C}_g(p) = 0$ for $2g > p$.

Now let l be the number of unpaired sides. To produce a (partial chord) diagram of the type in figure 45, we choose the l unpaired vertices from the $2p + l$ available slots. So we have a relation

$$\mathcal{N}_g(p,l) = \begin{pmatrix} 2p+l \\ l \end{pmatrix}\mathcal{C}_g(p) = \frac{(2p+l)!}{2p!\,l!}\mathcal{C}_g(p).$$

Substituting this in equation (56), we obtain

$$4p(p+1)\mathcal{N}_g(p,l) = 4(2p+l)(2p+l-1)\mathcal{N}_g(p-1,l)+ \\ (2p+l)(2p+l-1)(2p+l-2)(2p+l-3)\mathcal{N}_{g-1}(p-2,l).$$

We show the numbers $\mathcal{N}_g(p,l)$ computed for p up to 7 and $l = 2$ in table 6.

	$g=0$	$g=1$	$g=2$	$g=3$
$p=1$	6			
$p=2$	30	15		
$p=3$	140	280		
$p=4$	630	3150	945	
$p=5$	2772	27720	31878	
$p=6$	12012	210210	588588	135135
$p=7$	51480	1441440	7927920	6795360

Table 6: The numbers $\mathcal{N}_g(p,l)$ for $l = 2$

5.5 Topological characteristics of protein metastructures

We compute genera and numbers of boundary components for metastructures generated from 8913 selected proteins from PDB ([20]; see section 5.6 for details of the selection process), which are not α-only structures, and do not contain β-barrels or bifurcations in β-sheets. Figure 46 shows frequency distribution of actual proteins by their genera and numbers of boundary components.

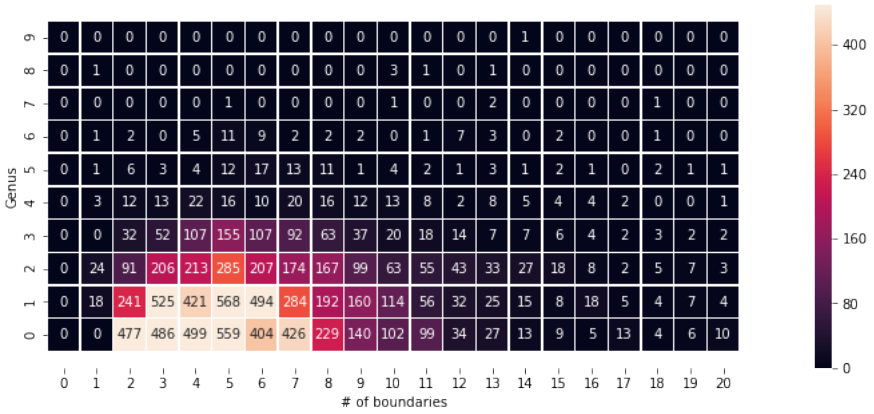


Figure 46: Frequency distribution (extract) of protein metastructures by genus and number of boundary components

The same distribution was also computed from the same number of simulated metastructures, produced as follows;

1. Reduced sequences were generated in the following manner.

   (a) The length was chosen randomly from log-normal distribution fitted to the distribution of PDB data.

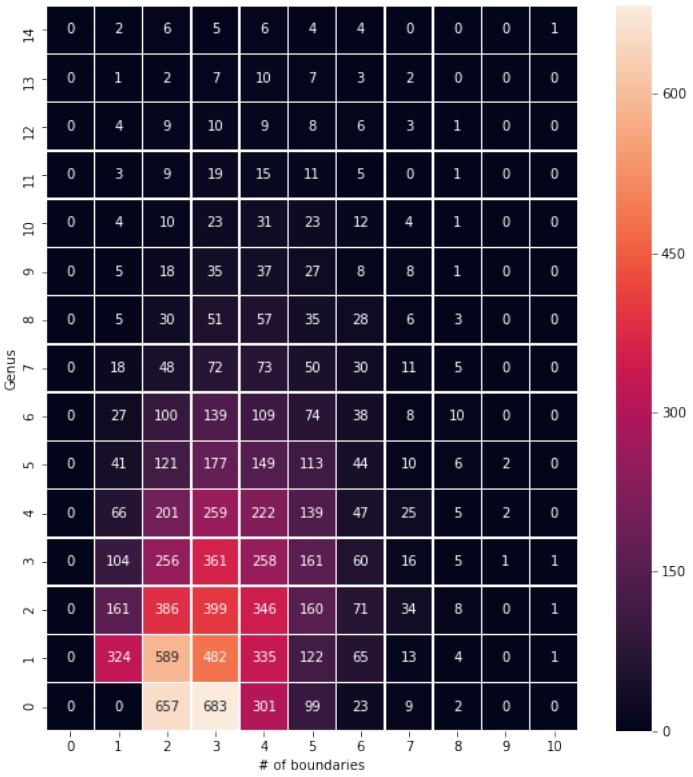


Figure 47: Frequency distribution (extract) of simulated protein metastructures by genus and number of boundary components

(b) Each pair of letters (1st and 2nd, 3rd and 4th, and so on) was given 50% chance of being "$\gamma\alpha$" and 50% chance of being "$\gamma\beta$". If the sequence has odd number of letters, the letter "γ" was attached at the end.

2. To each reduced sequence generated as above, a fatgraph structure was assigned as follows.

   (a) Let U be the set of letter βs in a given sequence, indexed with their positions in the sequence; $\beta_1, \beta_2, \ldots$. Then we partition U into a random number of subsets, each containing at least 2 elements.

   (b) For each subset U_i, choose a random ordering of β_is in the subset. This defines the ordering of strands in a beta-sheet.

   (c) For each ordered subset U_i with n_i elements, choose a random sequence of 1 and -1, of length n_i, but starting with 1. This sequence defines parallel/anti-parallel orientation of each strand with respect to the previous strand in a sheet.

We observe that the actual data tends to favour lower genera (and higher number of boundary components) compared to the simulated data (figure 47). This implies that metastructures whose associated surfaces have lower genera are favoured over those that result in high genera in the nature. Inspired by this observation, we will develop a method for prediction of β-sheet topology using the characteristics of the protein's associated surface in section 5.6.

For later use, we compute the distribution of the actual protein data by genus, number of boundary components and number of β-strands in the largest β-sheet. We call this the 3-dimensional genus-boundary distribution (see appendix B).

5.6 Dataset

The dataset used was based on HQ60 dataset in [73]. Here we give a brief summary of this dataset. PISCES [95] is a service that, among other things, creates subsets of sequences from PDB based on specified threshold for structure quality and sequence identity. For the HQ60 dataset, we use only X-ray structures, with a resolution threshold of 2.0Å, Rfac threshold of 0.2, and maximum sequence homology of 60%. This produced a list of 15548 proteins (as of March 2020). The hydrogen bonds are taken from the DSSP program [51], with the additional conditions [19];

$$\begin{aligned} &\text{HO-distance} < 2.7\text{Å} \\ &\text{angle(NHO)}, \text{angle(COH)} > 90^\circ. \end{aligned}$$

The secondary structures are also determined by DSSP, and they are recorded with three main secondary classes; [H]elix for H, G or I 8-state classes, [S]heet for E, and [C]oil for others. Thus we obtain, for each protein (of length n) in the dataset, a primary sequence $a_1 a_2 \cdots a_n$, where a_i is one of the 20 standard gene code amino acids, and a secondary structure sequence $b_1 b_2 \cdots b_n$, where $b_i = \alpha, \beta$, or γ. We superimpose these two sequences to obtain a *hybrid sequence* $c_1 c_2 \cdots c_n$, where $c_i = b_i$ if $b_i = \alpha$ or β, and $c_i = a_i$ otherwise. Together

with the information about hydrogen bonds, we are able to identify β-strands, their pairings and whether the pairing is parallel or anti-parallel (see section 5.2 for details). We will use the protein metastructures, not the extended metastructures, for purpose of the current analysis. This choice was made because it was not possible to obtain the extended metastructure information from the reference software we used for the analysis, Betapro [29], as well as for the ease of programming the model. For the analyses of metastructures, we are only interested in proteins containing β-sheets. Furthermore, proteins containing β-barrels, isolated strands and bifurcations in β-sheets are removed. This resulted in 8913 proteins. See table 7 for the number of proteins in each category.

α only	1471	(9.4%)
Bifurcation	3647	(23.5%)
β-barrel & Isolated strand	1517	(9.8%)
Accepted for analysis	8913	(57.3%)
Total (HQ60)	15548	

Table 7: Number of proteins filtered from the dataset.

5.7 Applications

We will now describe a series of experiments to attempt to utilise the topological characteristics of protein metastructures described in section 5.5.

5.7.1 Binary classification of candidate structures by their topology

200 proteins are randomly chosen for validation from the dataset, and the remaining 8713 proteins are used as the learning data. The idea is to use the learning data to decide the local configuration of β-strands, i.e. those strands, that are close to each other along the backbone. We then use the global topological data to decide the global configuration of the local blocks. We will now describe the first part of the method below. The aim is to first populate the pairing matrix $\mathbb{P}$ along the super- and sub-diagonals (i.e. the entries directly above and below the diagonal). We then repeat the procedure to populate the second entries above and below the diagonal, then the third entries, and so on. Pseudocode for populating the super- and sub-diagonals in the pairing matrix is given in algorithm 1.

1. From each protein in the validation data, we consider its hybrid sequence and extract segments between two β-strands.

2. For each extracted segment s, compute alignment score for all segments from the learning data using the Needleman-Wunsch algorithm[1] [66].

3. Let t be the segment from the learning data with the highest alignment score. The configuration of two strands at either end of segment t (whether they are paired by hydrogen bonds, and if so whether parallel or anti-parallel) determines the configuration of two strands at either end of s.

[1]For the substitution matrix we use blosum62 [46], extended by setting a match score with α or β to 4 and mismatch involving α or β to -4. See appendix A for more details.

4. Normalise the alignment score by dividing it by the score for perfect match, and record it in the appropriate entry in the pairing matrix $\mathbb{P}$. Specifically, if $p(s,t)$ is the alignment score for segments s and t, the normalised score $\tilde{p}(s,t)$ is given by $p(s,t)/p(s,s)$. Suppose s is the segment between i-th and $i+1$-th β-strands, and that they should be paired in the parallel configuration. Then set $\mathbb{P}_{(i,i+1)} = \tilde{p}(s,t)$.

5. If there is a tie for the highest alignment score, compute average of pairing scores (which is 1 if the two strands are paired, and 0 if not) among the highest-scoring segments. The two strands at either end of s are paired, if and only if the average is ≥ 0.5. The parallel/anti-parallel configuration of the two strands is determined similarly.

The above procedure allows us to populate $\mathbb{P}$ along the super- and sub-diagonals. We now repeat the procedure with s being a segment containing k β-strands, $k = 1, 2, 3, \ldots$, such that s is the segment between i-th and $i+k+1$-th β-strands. We do this to populate $\mathbb{P}$ up to d entries above and below the diagonal, where d is given by;

$$d = \begin{cases} 1 & \text{if } n < 7 \\ n - 5 & \text{if } 7 \leq n < 11 \\ 5 & \text{if } 11 \leq n. \end{cases}$$

Here the limit of 5 for d is forced by the fact that the learning data contains too few segments containing more than 4 β-strands. Furthermore, as the segments get longer it becomes harder to obtain high alignment scores, resulting in the chance of having $\mathbb{P}_{(i,j)} = 1$ being extremely small, when $|i-j| > 4$ (We were not able to get 1 in these cells in our tests). This is possibly related to the fact that the above method is essentially a method based on local data, and thus is not suited for predicting non-local configuration of β-strands. For that, another approach is needed which takes into account the global characteristics, which we will describe in the second part of the method. Before that, we need to translate the entries of the partial pairing matrix computed above, which are real numbers between 0 and 1, to either 0 or 1. We do this by changing the non-zero entries to 1, starting from the largest to the smallest. If, at any point, setting an entry to 1 results in a bifurcation or a β-barrel, the entry is set to 0 and we move onto the next largest entry (figure 48). For later use, we name this procedure `MakeBinary()`, which takes a (partial) matrix of pairing scores as an input and returns a (partial) pairing matrix.

We now have a partial pairing matrix, populated up to d entries above and below the diagonal, without bifurcations or barrels. We populate the remaining entries (the entries outside the d entries above and below the diagonal) by going through all possibilities, while avoiding bifurcations and β-barrels. We also require that the resulting matrix does not contain any isolated strand. The result is a number of candidate matrices, whose number depends on the partial pairing matrix computed in the first part of the method. We now construct a fatgraph from each candidate matrix, and compute its genus and number of boundary components, together with the number of strands in the largest sheet. We compare this data with the 3-dimensional genus-boundary distribution computed in section 5.5. By a layer in the 3-dimensional genus-boundary

Algorithm 1 Pseudocode for populating the first diagonal in the pairing matrix $\mathbb{P}$

Let $t_1, t_2, \ldots, t_m$ be the hybrid segments (consisting of some or all of 20 residues and α for helix segments) between two β-strands, extracted from all proteins in the learning dataset.
Let $s_1, s_2, \ldots, s_n$ be the hybrid segments between two β-strands in a given protein in the validation data.
Let $\mathbb{P}$ by an empty $n \times n$ matrix.
for s_i in $s_1, s_2, \ldots, s_n$ **do**
  for t_j in $t_1, t_2, \ldots, t_m$ **do**
    Compute alignment score $p(s_i, t_j)$.
  end for
  Let $\tilde{j} \in \{1, 2, \ldots, m\}$ such that $p(s_i, t_{\tilde{j}}) = \max_j\{p(s_i, t_j)\}$.
  if $\tilde{j}$ is uniquely determined **then**
    if The two segments at either ends of $t_{\tilde{j}}$ are paired **then**
      Set $\tilde{p_i} = p(s_i, t_{\tilde{j}})/p(s_i, s_i)$.
      if The two segments are paired in parallel configuration **then**
        Set $\mathbb{P}_{(i,i+1)} = \tilde{p_i}$ and $\mathbb{P}_{(i+1,i)} = 0$.
      else
        Set $\mathbb{P}_{(i+1,i)} = \tilde{p_i}$ and $\mathbb{P}_{(i,i+1)} = 0$.
      end if
    else
      Set $\mathbb{P}_{(i,i+1)} = \mathbb{P}_{(i+1,i)} = 0$.
    end if
  else
    Let $\tilde{j_1}, \tilde{j_2}, \ldots, \tilde{j_k}$ be such that $p(s_i, t_{\tilde{j_h}}) = \max_j\{p(s_i, t_j)\}$ for all $h \in \{1, 2, \ldots, k\}$.
    Set $\bar{q} = \frac{1}{k}\sum_h q_h$, where $q_h = 1$ if the two segments at either ends of $t_{\tilde{j_h}}$ are paired, $q_h = 0$ if not.
    if $\bar{q} \geq 0.5$ **then**
      Set $\tilde{p_i} = p(s_i, t_{\tilde{j}})/p(s_i, s_i)$.
      Let x be the number of $\tilde{j_h}$, where the two segments at either ends of are paired in parallel configuration. Let y be the number for anti-parallel configuration.
      if $x \geq y$ **then**
        Set $\mathbb{P}_{(i,i+1)} = \tilde{p_i}$ and $\mathbb{P}_{(i+1,i)} = 0$.
      else
        Set $\mathbb{P}_{(i+1,i)} = \tilde{p_i}$ and $\mathbb{P}_{(i,i+1)} = 0$.
      end if
    else
      Set $\mathbb{P}_{(i,i+1)} = \mathbb{P}_{(i+1,i)} = 0$.
    end if
  end if
end for

	1	2	3	4	5	6	7	8
1	0	0.95	0	0				
2	0	0	0	0.87	0			
3	0	0.98	0	0	0	0		
4	0	0	0	0	0	0	0	
5		0	0	0.72	0	0	0	0
6			0	0.23	0.46	0	0	0
7				0	0	0	0	0
8					0	0	0.60	0

	1	2	3	4	5	6	7	8
1	0	1	0	0				
2	0	0	0	0	0			
3	0	1	0	0	0	0		
4	0	0	0	0	0	0	0	
5		0	0	1	0	0	0	0
6			0	0	1	0	0	0
7				0	0	0	0	0
8					0	0	1	0

Figure 48: Construction of a partial pairing matrix (right) from a partial score matrix (left). We start with the highest alignment score and set the first two, 0.98 and 0.95 to 1. The third highest, 0.87, would result in bifurcation, so it is set to 0. The next three scores are set to 1, but the last non-zero entry, 0.23 will result in a barrel involving strands 4, 5, and 6, so it is set to 0. The resulting partial pairing matrix has three blocks, listed as a set of strands, (1,2,3), (4,5,6) and (7,8). Filling this matrix by either 0 or 1 would result in $2^{20} = 1048576$ different matrices, but the restrictions placed on pairing matrices means there are only 97 valid completions.

distribution, we mean the 2-dimensional distribution of genus and number of boundary components for a specific value of number of strands in the largest sheet. Let g, n, l denote the genus, the number of boundary components and the number of strands in the largest sheet. Let $f(g, n, l)$ be the frequency of the cell (g, n, l) in the 3-dimensional genus-boundary distribution. We define the topology score $s_{\text{topo}}(\tau)$ for a metastructure τ with genus g, n boundary components and l strands in the largest sheet, by

$$s_{\text{topo}}(\tau) = \frac{f(g, n, l)}{T_l},$$

where T_l is the sum of frequencies for the lth layer. For a cutoff value $v \in (0, 1)$, a candidate metastructure τ is accepted, if $s_{\text{topo}}(\tau) \geq v$, and rejected if $s_{\text{topo}}(\tau) < v$. We also compute accuracy of each candidate structure, and look at the relationship between accuracy and acceptance of candidate structures.

5.7.2 Metastructure prediction by sequence alignment and topology

The method described in section 5.7.1 was modified to provide a single, "best candidate" metastructure. The modification was made such that instead of classifying candidate metastructures as either accepted or rejected, a weighted sum of all candidate pairing matrices was produced, with weight given by the 3-dimensional genus-boundary distribution. More precisely, suppose a candidate pairing matrix $\mathbb{P}$ results in a structure with genus g, n boundary components and l strands in the largest sheet. Let $f(g, n, l)$ be the frequency of the cell

(g, n, l) in the 3-dimensional distribution, and T_l be the sum of frequencies for the lth layer, as before. Then our final score matrix $\hat{\mathbb{P}}_{\text{score}}$ is given by

$$\hat{\mathbb{P}}_{\text{score}} = \sum_{\mathbb{P}} \frac{f(g,n,l)}{T_l} \mathbb{P},$$

where the sum is over all candidate pairing matrices for a protein. The final pairing matrix $\hat{\mathbb{P}}$ is computed from $\hat{\mathbb{P}}_{\text{score}}$ as before. A pseudocode for this procedure is shown in algorithm 2.

Algorithm 2 Pseudocode for computation of prediction pairing matrix $\hat{\mathbb{P}}$.

Let $\mathbb{P}_{\text{partial}}$ be a given partial pairing matrix.
Let $\hat{\mathbb{P}}_{\text{score}}$ be a zero matrix of the same size as $\mathbb{P}_{\text{partial}}$.
for all Completion $\mathbb{P}$ of $\mathbb{P}_{\text{partial}}$ **do**
  if $\mathbb{P}$ contains a barrel, a bifurcation or an isolated strand **then**
    Move to next completion
  end if
  Compute genus g, number of boundary components n, and size of the largest sheet l for the metastructure corresponding to $\mathbb{P}$.
  Find the frequency of the cell (g, n, l) and the sum of frequencies for the lth layer T_l.
  $\hat{\mathbb{P}}_{\text{score}} = \hat{\mathbb{P}}_{\text{score}} + \frac{f(g,n,l)}{T_l}\mathbb{P}$
end for
Set $\hat{\mathbb{P}} = \textsc{MakeBinary}(\hat{\mathbb{P}}_{\text{score}})$

5.7.3 Metastructure prediction by Betapro and topology

Betapro is a computer programme for predicting β-sheet topology using recurrent neural network (RNN) [29]. It takes a primary structure sequence as input, or a primary and secondary structure sequences, if the secondary structure is available from other sources. The output is a score matrix, where the entries are not restricted to $(0, 1)$, but positive real numbers computed as a sum of pseudoenergy for each residue pair in a β-strand pairing. The reported performances of Betapro are 0.54 for Recall and 0.59 for Precision [29].

In order to predict protein metastructure, we run Betapro using the primary and secondary structure sequences as input. From the output score matrix, we choose m entries with the highest scores, where m equals 4% of the number of entries in the score matrix, excluding the main diagonal. The entries that result in a bifurcation or a barrel, are ignored. The chosen entries are considered as β-strand pairings, and they are set to 1 in the partial pairing matrix. Next, all valid (i.e. avoiding isolated strands, bifurcations and barrels) completions of the partial pairing matrix are generated. Each completion is given two scores, one based on Betapro score matrix, and the other based on the genus-boundary distribution. The first, s_{bp}, is the sum of all scores in Betapro score matrix, where there is 1 in the pairing matrix. The second, s_{topo}, is given by $f(g, n, l)/T_l$, where g, n, l is the genus, the number of boundary components, and size of the largest sheet, as before. Our prediction is the structure with the highest combined score,

$$\hat{s} = a s_{\text{bp}} + b s_{\text{topo}}, \tag{57}$$

where $a, b \in [0, 1]$ with $a + b = 1$. The corresponding pseudocode is shown in algorithm 3.

Algorithm 3 Pseudocode for computation of prediction pairing matrix $\hat{\mathbb{P}}$ from Betapro score matrix $\mathbb{P}_{\text{bp}}$.

Let $\mathbb{P}_{\text{bp}}$ be the pairing score matrix produced by Betapro.
Let $\mathbb{P}_{\text{partial}}$ be an empty matrix of the same size as $\mathbb{P}_{\text{bp}}$.
Order the entries in $\mathbb{P}_{\text{bp}}$ from largest to smallest.
Set $c = 0$.
while $c \leq m$ **do**
    Let (i, j) be the index for the first element in the ordered list of entries in $\mathbb{P}_{\text{bp}}$.
    Set $\mathbb{P}_{\text{partial}(i,j)} = 1$.
    if $\mathbb{P}_{\text{partial}}$ results in a barrel or a bifurcation **then**
        Set $\mathbb{P}_{\text{partial}(i,j)} = 0$.
        $c = c - 1$.
    end if
    Remove the first element from the ordered list of entries in $\mathbb{P}_{\text{bp}}$.
    $c = c + 1$
end while
for all Completion $\mathbb{P}$ of $\mathbb{P}_{\text{partial}}$ **do**
    if $\mathbb{P}$ contains a barrel, a bifurcation or an isolated strand **then**
        Move to next completion
    end if
    Compute genus g, number of boundary components n, and size of the largest sheet l for the metastructure corresponding to $\mathbb{P}$.
    Find the frequency of the cell (g, n, l) and the sum of frequencies for the lth layer T_l.
    Set $s_{\text{topo}}(\mathbb{P}) = \frac{f(g,n,l)}{T_l}$.
    Set $\mathbb{P}_{\text{score}} = \mathbb{P} \dot{\times} \mathbb{P}_{\text{bp}}$, where $\dot{\times}$ denotes the entry-wise multiplication.
    Set $s_{\text{bp}}(\mathbb{P}) = \sum_{i,j} \mathbb{P}_{\text{score}(i,j)}$.
    Set $\hat{s}(\mathbb{P}) = a s_{\text{bp}}(\mathbb{P}) + b s_{\text{topo}}(\mathbb{P})$.
end for
Set $\hat{\mathbb{P}}$ to be the completion $\mathbb{P}'$, such that $\hat{s}(\mathbb{P}') = \max\{\hat{s}(\mathbb{P}) | \mathbb{P} \text{ is a completion of } \mathbb{P}_{\text{partial}}\}$.

5.7.4 Results

Some of the larger proteins in the 200 test proteins could not be analysed using the method described, as there were too many possible ways to complete the pairing matrix. We therefore limit the analysis to the 189 proteins containing up to 20 β-strands. Their frequency distribution by the number of β-strands is shown in figure 49.

The algorithm from section 5.7.1 produced 66,789,038 candidate structures in total, but there are significant variations in the number of candidate structures per protein (figure 50), as the possible number of candidates also depends on the partial structure determined using alignment of the α/γ segments between β-strands. In the current analysis, one protein (1H16A) accounted for

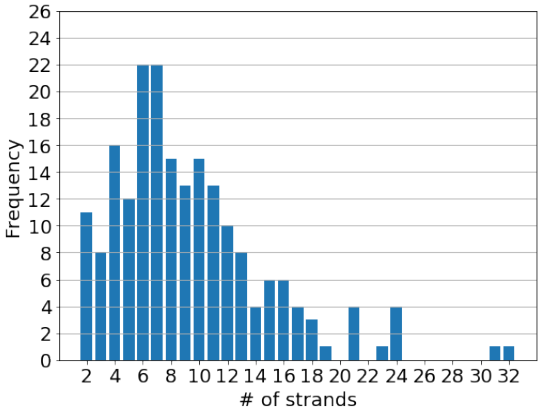


Figure 49: Frequency distribution of 200 proteins by the number of strands

45,714,440 candidate structures, representing 68% of the total number. Note, although some of these numbers are large, they still represent a significant reduction from the theoretically possible number of candidate structures, which is given by $n! \cdot 2^{n-2}$ for a protein with n strands, when considering only those structures with a single sheet. Naturally the numbers are even larger when considering multiple-sheet structures. We list the first few terms in table 8.

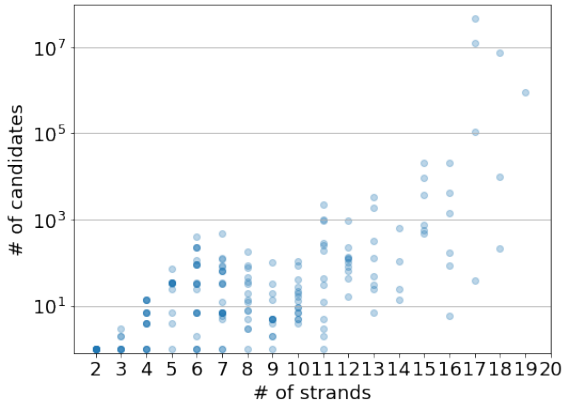


Figure 50: Number of candidate structures per protein, filtered by the number of strands. Note the log scale. There are large variations in the number of candidates among the proteins with the same number of strands.

The topology filter, depending on the cutoff value and the number of strands, further reduces the number of candidate structures (figure 51). Upon considering the balance between the ability to reduce the number of candidate structures and still retain high quality candidate structures, we decided to use the cutoff s_{topo} value of 0.02 for the subsequent analysis. The actual number of accepted candidate structures are shown in figure 52. As we also can see from figure 51,

Strands	Number of structures	
	Single sheet	Multiple sheets
2	2	2
3	12	12
4	96	108
5	960	1200
6	11520	15960
7	161280	246960

Table 8: The number of theoretically possible structures for a protein with n strands.

the topology filter is very effective at reducing the number of structures for proteins with larger number of strands (i.e. large number of candidates). In the current analysis, there were only 4 out of 189 proteins, where the number of accepted candidates were more than 5,000. For these the topology filter reduced the number of candidates by 92-95%. When using any positive cutoff value for such a filter, there is a chance that no candidate structure for a protein is accepted. If it happens, we reduce the cutoff value only for the proteins with no accepted candidate structure, until one or more candidate structures are accepted. In the current analysis, the cutoff values were reduced by 0.005 down to 0.005. If, at the end of this iteration, we have proteins with no accepted structure, we randomly select one candidate structure for acceptance. This procedure, however, was not necessary for the current analysis, and all proteins had at least one candidate structure accepted at the cutoff value of 0.02.

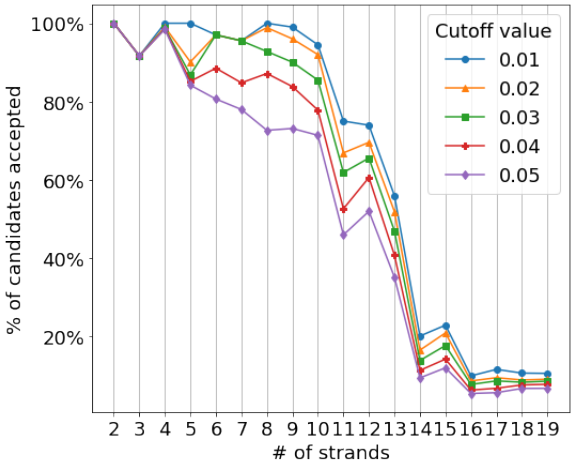


Figure 51: Percentages of accepted structures by cutoff values and the number of strands.

In order to examine how well our topological filter distinguishes between "good" and "bad" candidates, we investigate how the rate of acceptance changes for "good" and "bad" candidate structures. Precision and Recall are two measures often used for judging quality of predicted protein structures. They are

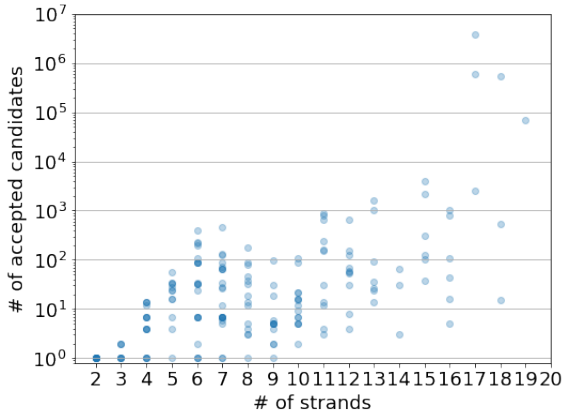


Figure 52: Number of accepted candidate structures per protein, filtered by the number of strands. Note the log scale. Compared to figure 50, the numbers are significantly lower where there are large number ($> 10^4$) of candidate structures.

# candidates	# accepted	% accepted
45714440	3718300	8.1%
12528000	586277	4.7%
7445742	536820	7.2%
904798	69232	7.7%

Table 9: The numbers and percentages of accepted structures for the four proteins with most candidate structures. The topology filter rejects more than 90% of candidates.

given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

where TP, FP and FN stand for the number of true positive, false positive and false negative strand pairings.

For a given quality measure Q (=Precision or Recall), we divide the candidate structures into three classes; low quality (structures with $Q < 0.6$), medium quality ($0.6 \leq Q < 0.9$), and high quality ($0.9 \leq Q$). We then compute the acceptance rate for each class. The results are shown in table 10. We see that for Precision, the acceptance rate for high quality structures is ten times as high as either low or medium quality structures. For Recall the difference is smaller but still higher for high quality structures.

If we exclude the four proteins with more than 5000 candidate structures, which together account for 99.7% of all candidate structures, the differences are greater (table 11), particulary for Recall.

Metastructure prediction by sequence alignment and topology (section 5.7.2) and by Betapro and topology (section 5.7.3) were performed on the same set

Quality	Precision		Recall	
Low	7.38%	(4850914/65738106)	7.42%	(4819915/64953828)
Medium	7.88%	(82780/1050828)	6.19%	(112857/1824369)
High	89.42%	(93/104)	9.36%	(1015/10841)

Table 10: Acceptance rate and number of acceptance (in parentheses, {# accepted}/{#candidates}) by quality classes.

Quality	Precision		Recall	
Low	15.18%	(13136/86540)	21.27%	(11729/55150)
Medium	9.07%	(9929/109416)	7.86%	(11042/140460)
High	91.18%	(93/102)	86.38%	(387/448)

Table 11: Acceptance rate and number of acceptance (in parentheses, {# accepted}/{#candidates}) by quality classes, excluding four proteins with > 5000 candidate structures.

of proteins. The average Precision and Recall for the predictions are shown in table 12. Different values of a in 57 only had a very small effect (< 0.005) on Precision or Recall values. We also applied logarithm to the strand pairing scores from Betapro and used them in the algorithm, which resulted in an increase in Precision but a (smaller) decline in Recall (table 12) This change was seen across different number of strands (figure 53). To investigate the effect of the number of selected pairings before computing completions, we ran the algorithm using 4,5,6% for pre-selection, together with the "fewest possible" pre-selections, which is the number where a computation is possible within a reasonable amount of time (24 hours). The number p of pre-selected pairs for a protein with n strands was;

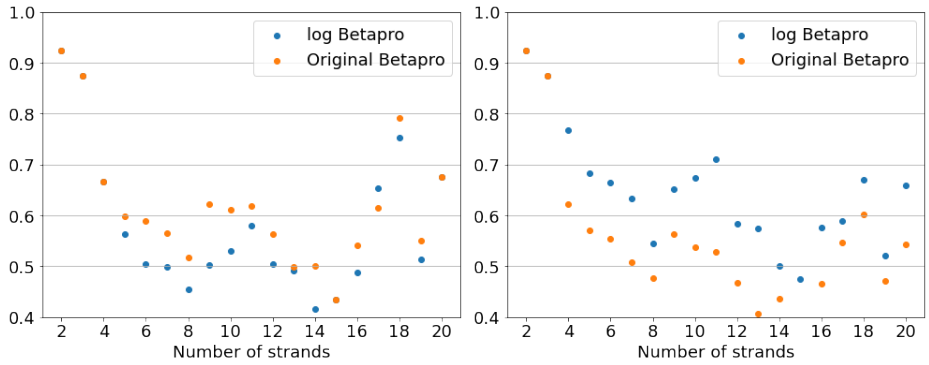
$$p = \begin{cases} 0 & \text{if } n \leq 8 \\ n-8 & \text{if } 9 \leq n \leq 11 \\ n-7 & \text{if } 12 \leq n \leq 20 \end{cases}$$

	by alignment	by Betapro		by logBetapro
		a=0.1	a=1	
Precision	0.36	0.56	0.56	0.67
Recall	0.36	0.62	0.62	0.57

Table 12: Average Precision and Recall for two metastructure prediction methods.

5.8 Discussion

The difference in the distributions of the genera and the numbers of boundary components from the actual (figure 46) and simulated data (figure 47) indicate that the folding of β-sheets is not a completely random process. Indeed, it does appear that an increase in genus is costly and a structure that has lower genus



(a) Average Recall (b) Average Precision

Figure 53: Average Recall (a) and Precision (b) by number of strands, Betapro and logBetapro scores.

	Fewest	4%	5%	6%
Precision	0.567	0.567	0.572	0.577
Recall	0.667	0.667	0.659	0.650

Table 13: Average Precision and Recall for different levels of pre-selection.

is favoured over one with higher genus. This observation agrees with previous studies, which do not look at genus of β-sheets, but finds that certain β-sheet structures, many of which correspond to an increase in genus, are absent or very rare in proteins [80, 99]. The result of our binary classification analysis (section 5.7.1) agrees with this observation. Even though the result is skewed by a highly uneven distribution of the number of candidate structures per protein, and the response of acceptance rate for an increase in quality is not linear, it does appear that the topology of protein metastructure captures some information about the native structure. Extending this result to prediction of metastructures proved more challenging. We did achieve a result comparable to that reported for Betapro when using strand-pairing scores as is, which was improved to be better than Betapro with an application of logarithm to the pairing scores. This is likely to be because the unprocessed Betapro scores are strictly greater than zero, thus encouraging formation of larger sheets in order to maximise the final score $\hat{s}$, even though the contribution from the topology score s_{topo} should, to some extent, prevent the formation of sheets that are too large and topologically complex. By applying logarithm to the Betapro scores, we encourage fewer pairings (and thus discourage large sheets), which resulted in improved Precision. We were, however, not able to outperform the figures reported by other, more recent studies (table 14). The structure of the BCov and BetaProbe programs meant that it was not possible to combine them with our method in a similar manner to section 5.7.3. It would be interesting to see if we can improve the results of Top-DBS program by combining with our method. Unfortunately the program code for Top-DBS was not available for inspection.

One of the reasons for why the results from our study could not match those

Program	Precision	Recall
Betapro [29]	0.59	0.54
BCov [81]	0.60	0.62
BetaProbe [34]	0.67	0.70
Top-DBS [31]	0.75	0.78
Current Study	0.67	0.57

Table 14: Comparison of Precision and Recall values for prediction of β-sheet topology.

from more recent studies may be that the topology filter, in its current form, is too coarse. Suppose we have a protein with three β-strands. There are 12 different protein metastructure configurations possible, but 8 of them have genus 0 and 3 boundary components, with the rest having genus 1 and 1 boundary component. This suggests a "finer" filter, which can distinguish between the structures having the same genus and number of boundary components (and maximum sheet size), may be able to produce a better result. However, with the size of the currently available dataset, making the filter finer would result in the frequency in each cell being too small for looking at the distribution of genera or numbers of boundary components (or some other topological data).

The term β-sheet topology is commonly used to describe the configuration of β-strands in a β-sheet. However, to our knowledge, it has not been studied in relation to topological invariants. We have shown in this project that the topological invariants such as genus and the number of boundary components can describe certain aspects of β-sheet topology of proteins, and how they might be used in prediction of β-sheet topologies. In the following chapters we will investigate how topology of an entire protein may be used to study different aspects of the protein folding problem.

6 Protein fatgraph and GDT

6.1 CASP and GDT

CASP: Critical Assessment of protein Structure Prediction is a large-scale, biannual experiment held since 1994, with an aim to improve the science of protein structure prediction [62, 58]. At the start of each round a set of primary sequences for proteins, whose structural data (also called the target structures) are kept secret, is published. The participating research groups makes structural predictions based on the published primary sequences, and submit their predictions for assessment. The submitted structures are assessed for accuracy using a number of metrics including GDT (global distance test) [57]. GDT is a metric first proposed in [98] as an improvement to another metric, the overall root mean square deviation (RMSD). RMSD is given by

$$\text{RMSD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\delta_i^2},$$

where n is the number of residues in the protein being assessed, and δ_i is the deviation in the positions of i'th C^α atoms in the two structures being compared. One can see from this definition, that RMSD can be sensitive to large deviations in a small, limited region of the protein structures in assessing their similarity. Note in Biology, the term RMSD is sometimes used to mean the above quantity computed specifically for the optimal alignment between two structures. Here we use the term more generally for any alignment between two structures with the same primary structure. GDT attempts to address this issue by providing more global measure of similarity. It is computed as follows;

We start by the representation of a protein as a sequence of C^α's, $A = (a_1, a_2, \ldots, a_n)$, each $a_i \in \mathbb{R}^3$, representing the position of a residue (C^α atom). Let $D = (d_1, \ldots, d_n)$ be a candidate protein and $T = (t_1, \ldots, t_n)$ be the target. Let $D^0 = (d_1, d_2, d_3)$ be the first three-residue segment of the candidate D and $T^0 = (t_1, t_2, t_3)$ be the corresponding segment of the target T.

a Obtain a transform (translation & rotation) R^0, which minimises RMSD between $R^0D^0 := (R^0d_1, R^0d_2, R^0d_3)$ and T^0, where $\text{RMSD} = \sqrt{(\sum_1^n \|R^0d_i - t_i\|^2)/n}$, with $n = 3$ in this case. Kabsch algorithm [50] is a widely used algorithm for obtaining the RMSD-minimising transform.

b Apply R^0 to D to obtain $R^0D := (R^0d_1, \ldots, R^0d_n)$.

c Measure the distance between each pair of residues, $\|t_i - R^0d_i\|$, to identify the subset J^0 of $I := \{1, 2, 3, \ldots, n\}$, given by $J^0 := \{i \in I : \|t_i - R^0d_i\| > x\}$, where x is a given distance cutoff.

d Let D^1 be the subset of D, obtained by removing from D all d_i with $i \in J^0$. Similarly we obtain a subsequence T^1 of T. We then obtain the RMSD-minimising transform R^1 for D^1 and T^1.

e We apply R^1 to D and repeat the procedure to obtain R^2. The algorithm terminates when $J^i = J^{i-1}$.

In this way we obtain the largest subset D_1 of D that can be superimposed onto T within the given cut-off distance x, for the seed segment $D^0 = (d_1, d_2, d_3)$. Let I_1 be the index set corresponding to this largest subset of D, i.e. $I_1 := \{i \in I : d_i \in D_1\}$. The entire procedure is then repeated for each three-residue segment $(d_i, d_{i+1}, d_{i+2}), i = 1, \ldots, n-2$, to obtain subsets I_i of I. Finally, we choose the largest subset $I_{\max}$ among I_i's, and report GDT for cut-off value x;

$$\mathrm{GDT} = \mathrm{GDT}_x = |I_{\max}| / |I| ,$$

where $|I|$ is the cardinality of the set I. Note here, the dependence on the cutoff value x. The metric used in CASP, called GDT_TS, is an average of GDTs for four different values of x [57]. It is given by;

$$\mathrm{GDT\_TS} = (\mathrm{GDT}_1 + \mathrm{GDT}_2 + \mathrm{GDT}_4 + \mathrm{GDT}_8)/4.$$

One of the experiments in CASP, performed since CASP7, is the model quality assessment, where the participants are asked to estimate the quality of all submitted models [57]. Inspired by this experiment, we investigated to what extent our protein model can be used to select the candidate with the best GDT_TS scores of CASP structures. We investigated two different methods; one is a linear regression where independent variables are certain similarity scores computed from the protein model, while the other attempts to follow the algorithm for computing GDT_TS, but with only the protein's topological information (from our protein model) as inputs. Our methods are not intended as an attempt for the model quality assessment. Indeed, both methods require the target structure's fatgraph model as part of the input data, which is not available in CASP. Rather, they are intended as an investigation into the usefulness of protein topology in the model quality assessment. Ultimately, it is hoped that a method will be developed which is able to predict a protein's fatgraph model with high accuracy, so that the information therein can be used for the model quality assessment.

In CASP, participating models are assessed by the difference between the GDT_TS score of the candidate identified by the model as the best, and the best candidate. This difference is also called ΔGDT. The assessment is usually performed for the targets, for which at least one candidate had a GDT_TS score of over 40. In CASP10, on which this analysis is based on, the best performing predictor had an average ΔGDT of 4.6 GDT_TS points, and it was able to identify a candidate with $\Delta\mathrm{GDT} < 2$ for approximately 33% of the time [55]. If we allowed for $\Delta\mathrm{GDT} < 10$, the figure was 90%. In CASP12, which is the latest round that reports this metric, the best performing predictor had an average ΔGDT of 5.0 GDT_TS points. It should however be noted that in CASP12, unlike in CASP10, one distinguished between so-called clustering methods and single-model methods, and the figure of 5.0 GDT_TS points is for the best single-model method. The best performing predictor was able to identify a candidate with $\Delta\mathrm{GDT} < 2$ for 40% of the time [56].

6.2 Dataset

The dataset consists of 91 target proteins and the submitted candidate structures from CASP10 [61]. The size of proteins, measured in the number of

residues, ranged from 33 to 770 (figure 54). The range of the number of candidate structures per protein was from 228 to 584 (Participants are allowed to submit more than one candidate structure). The histogram for the number of candidates per target is given in figure 55. It appears the distribution has two peaks, and that the targets with fewer submitted candidates had higher quality predictions. Indeed, there were 47 targets with number of candidates less than 400, of which all 47 had a candidate with GDT_TS higher than 40, while of the 44 targets with number of candidates more than 400, only 27 had a candidate with GDT higher than 40. This may be a result of there being fewer submissions for the easier targets, as participants are more certain of their prediction, while for the more difficult targets, the number of submissions were larger, reflecting increased uncertainties.

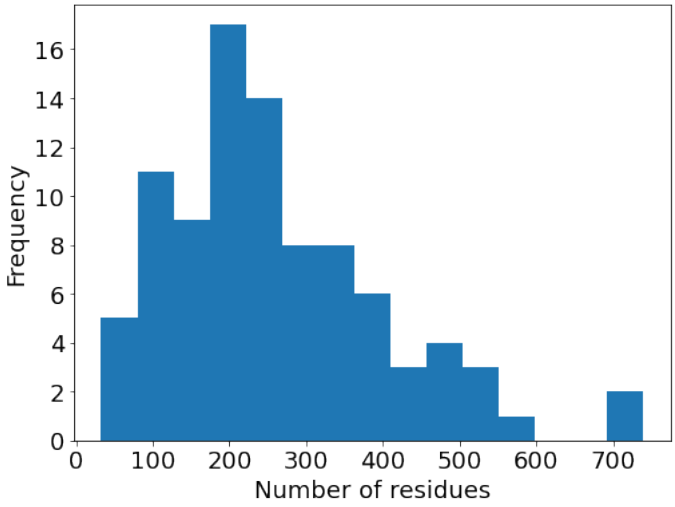


Figure 54: Frequency of target structures by length

The data was processed to obtain information about hydrogen bonds, in a manner similar to the method described in section 5.6. The hydrogen bonds were determined by DSSP program [51], with the additional conditions [19];

$$\text{HO-distance} < 2.7\text{Å}$$
$$\text{angle(NHO)}, \text{angle(COH)} > 90°.$$

6.3 Linear regression based on the protein fatgraph model

The first method is a linear regression on the similarity scores which we compute based on the hydrogen bonds in the candidate and target structures. Each hydrogen bond is identified by the position of its donor- and acceptor atoms, so each bond can be expressed as a 2-tuple of integers (p, q), where the donor is the p'th atom along the backbone and the acceptor the q'th.

The first of our similarity scores is the proportion of the bonds, which are correctly identified in the candidate structure. In other words, if T, C are the sets of hydrogen bonds respectively in the target structure and in the candidate

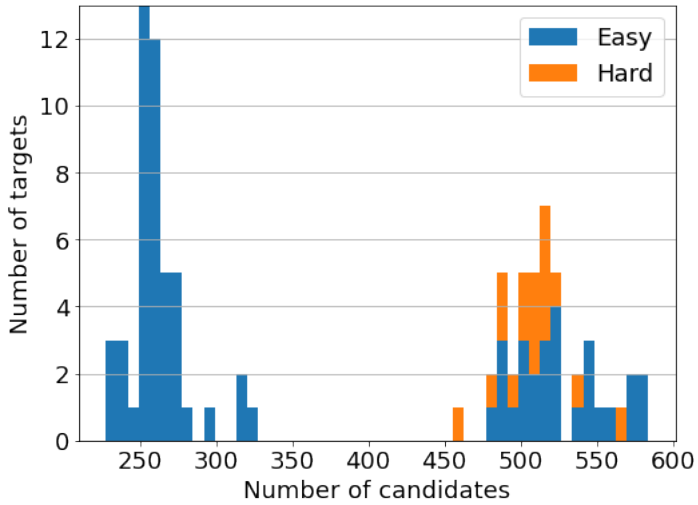


Figure 55: Number of candidate structures for 'Easy' (the best candidate has GDT_TS>40) and 'Hard' (the best candidate has GDT_TS≤40) targets.

structure, then the first score P is defined as;

$$P = \frac{\#(T \cap C)}{\#T},$$

where we use the fact that for two bonds (p, q) and (p', q'), $(p, q) = (p', q')$ iff $p = p'$ and $q = q'$.

The second similarity score S_n depends on a parameter $n \in \mathbb{N}$. For a non-negative integer $x \in \mathbb{Z}$, define

$$f_n(x) = \begin{cases} 1 - x/n & \text{if } x \leq 2n \\ -1 & \text{otherwise} \end{cases}.$$

For a bond $(p, q) \in C$, set

$$s_C((p, q)) = \max\left\{ f_n(|p - p'|) + f_n(|q - q'|) \mid (p', q') \in T \setminus C \right\}.$$

Similarly for $(p, q) \in T$, set

$$s_T((p, q)) = \max\left\{ f_n(|p - p'|) + f_n(|q - q'|) \mid (p', q') \in C \setminus T \right\}.$$

S_n is then given by

$$S_n = \frac{\sum_{(p,q) \in C \setminus T} s_C((p, q)) + \sum_{(p',q') \in T \setminus C} s_T((p', q'))}{\#((T \setminus C) \cup (C \setminus T))}.$$

So for a given candidate structure, we can compute S_n for different n's.

Having calculated P and S_n, $n \in I$, where I is some subset of $\mathbb{N}$, for all candidate structures, we perform a linear regression with P, S_n as independent

variables and GDT as the dependent variable. After testing for various subset $I \subset \mathbb{N}$, we found that setting $I = \{6, 8, 10\}$ gave the best results. Using $I = \{6, 8, 10\}$, we selected the best candidate structure for each target. The average difference in the GDT's of the best candidate and our prediction (ΔGDT) was 5.52 (figure 56). We were able to identify a candidate with $\Delta\text{GDT} < 2$ for 31% of the targets.

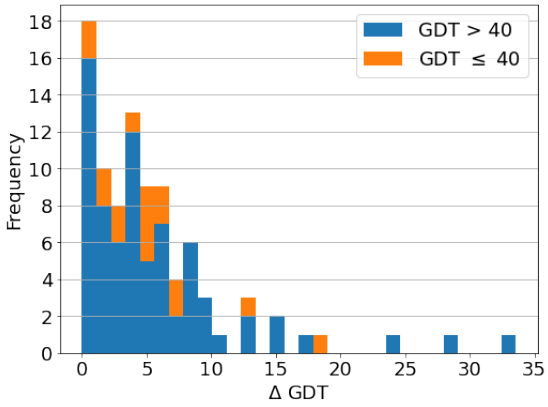


Figure 56: Frequency distribution of ΔGDT for 91 target structures, by linear regression method.

6.4 GDT-like algorithm based on the protein fatgraph model

Another approach for predicting GDT score based on our protein model is to mimic the GDT algorithm, but based on protein graphs, i.e. based on information about the protein's hydrogen bonds, but not its geometric structure.

Let T be the graph of the target protein, with vertices $\{v_1, \ldots, v_l\}$ representing the residues, ordered along the backbone, and edges $\{e_1, \ldots, e_m\}$ representing primary and secondary bonds. Similarly, let D be the graph of the candidate protein, with vertices $\{w_1, \ldots, w_l\}$ representing the residues, ordered along the backbone, and edges $\{f_1, \ldots, f_n\}$ representing primary and secondary bonds. Note $l = \#$ of vertices in $T (= \#$ of vertices in $D)$. Let $r \in (0, \infty)$. The idea is to start with small subgraphs of D and T (corresponding to the same backbone segment), and to "grow" them incrementally, until the difference between the subgraphs is over a pre-determined limit value. We repeat this for different initial subgraphs, and determine the maximum subgraph of D, whose difference from the corresponding subgraph of T is below the limit value. We present the pseudocode for the algorithm in algorithm 4.

The pseudocode in algorithm 4 can be roughly described as follows.

1. Set $i = 1$.

2. We select a subgraph $D_{\text{sub}}(i)$ of D, consisting of three vertices $\{w_i, w_{i+1}, w_{i+2}\}$ starting from the ith position and the edges connecting them. Select a subgraph $T_{\text{sub}}(i)$ of T in the same way.

Algorithm 4 Pseudocode for GDT-like algorithm

for i in $\{1, \ldots, l-2\}$ **do**
Let $D_{\text{sub}}(i)$ be the subgraph of D obtained by taking three vertices $\{w_i, w_{i+1}, w_{i+2}\}$ and the edges connecting them in D
Let $T_{\text{sub}}(i)$ be the subgraph of T obtained by taking three vertices $\{v_i, v_{i+1}, v_{i+2}\}$ and the edges connecting them in T
Compute the distance measure $d(i) = d(T_{\text{sub}}(i), D_{\text{sub}}(i))$
if $d(i) \geq r$ **then** #initial seed segment is already over the limit value
Continue to next i
end if
while True **do**
Let $T_{\text{sub}}2(i)$ be the subgraph of T obtained by taking $T_{\text{sub}}(i)$ together with all edges connected to the vertices in $T_{\text{sub}}(i)$, and the end-vertices of these edges (i.e. "grow" the subgraph by 1 edge+vertex pair)
Let $D_{\text{sub}}2(i)$ be the subgraph of D obtained in the same manner
Compute the distance measure $d(i) = d(T_{\text{sub}}2(i), D_{\text{sub}}2(i))$
if $d(i) \geq r$ **then**
Break
end if
Set $T_{\text{sub}}(i) = T_{\text{sub}}2(i)$
Set $D_{\text{sub}}(i) = D_{\text{sub}}2(i)$
if $T_{\text{sub}}(i) == T$ **then** #We can't grow $T_{\text{sub}}(i)$ any more
Break
end if
end while
end for
$D_{\text{sub\_max}} = \max\left\{D_{\text{sub}}(i) \middle| i \in \{1, \ldots, l-2\}\right\}$
Score $= 100 \cdot$ # of vertices in $D_{\text{sub\_max}}$ /# of vertices in D

3. Compute the distance measure $d(i) = d(T_{\text{sub}}(i), D_{\text{sub}}(i))$.

4. If $d(i) \geq r$, where r is the pre-determined limit, the initial segment is already over the limit value. Increment i by 1, go to 2.

5. If $d(i) < r$, "grow" the subgraph $D_{\text{sub}}(i)$ by one edge/vertex pair by selecting all edges connected to the vertices in $D_{\text{sub}}(i)$, and the end-vertices of these edges. Call the selected edges and vertices, together with $D_{\text{sub}}(i)$, $D_{\text{sub}}2(i)$. Select $T_{\text{sub}}2(i)$ from T in the same way.

6. Compute the distance measure $d(i) = d(T_{\text{sub}}2(i), D_{\text{sub}}2(i))$.

7. If $d(i) < r$, "grow" the subgraphs by one edge/vertex pair again and compute $d(i)$. If $d(i) \geq r$, move to the next starting segments by incrementing i by 1.

8. If $D_{\text{sub}}(i) == D$, we have the entire graph under the limit value.

9. After going through all starting segments, we have a set $S = \left\{ D_{\text{sub}}(i) \middle| i \in \{1, \ldots, l-2\} \right\}$ of maximal $D_{\text{sub}}(i)$'s. Select the longest $D_{\text{sub}}(i)$ in S, which we call $D_{\text{sub\_max}}$.

10. Score $= 100 \times \frac{\#\text{ of vertices in } D_{\text{sub\_max}}}{\#\text{ of vertices in } D}$.

For the current analysis we define the distance function d by

$$d(A, B) = \#\left((\mathcal{E}(A) \setminus \mathcal{E}(B)) \cup (\mathcal{E}(B) \setminus \mathcal{E}(A)) \right),$$

where $\mathcal{E}(A)$ is the set of edges in the graph A. Using this distance function, we predicted the best candidate structure for each target by selecting the structure with the highest score, which we call *graph-GDT*. The distribution of ΔGDT is shown in figure 57. The average ΔGDT for all targets was 19.1, and we were able to identify a candidate with $\Delta\text{GDT} < 2$ in 3.2% of the targets. We also compute the proportions of predictions with with $\Delta\text{GDT} < 2$, $2 \leq \Delta\text{GDT} \leq 10$ and $\Delta\text{GDT} > 10$ for both methods and compare them with the results from CASP10 [55] and CASP12 [56] (figure 58).

6.5 Discussion

Even though we were note able to outperform the best performing predictors, we were able to show that a similarity in proteins' geometric structures can be explained by similarity in their topology to a certain extent. The algorithm used in the second method was probably too naive, in that it does not differentiate between the backbone and hydrogen bond edges, nor the residue information given in the form of primary sequence. It may be possible to develop the algorithm further by utilising these data. The fact that it was nonetheless able to achieve the average ΔGDT of 19.1 can be seen as an indication of the magnitude of the influence, the topology of proteins has on their geometric structure.

There are broadly two types of methods used in CASP model accuracy estimation. Consensus, or clustering methods take multiple candidate structures as input and tries to identify a structure, that is the "best match" for the input structures according to some criteria. Single-model methods, on the other hand, takes a single candidate structure as an input and tries to estimate its

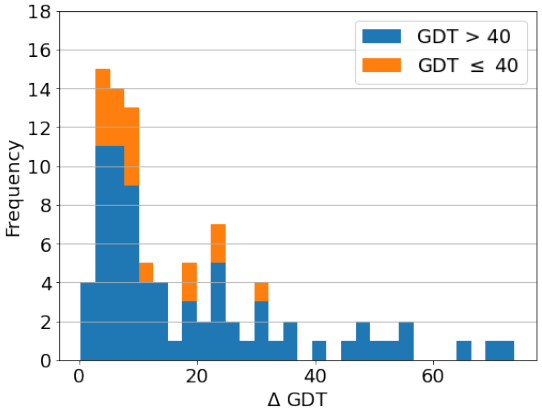


Figure 57: Frequency distribution of ΔGDT for 91 target structures, predicted using graph-GDT.

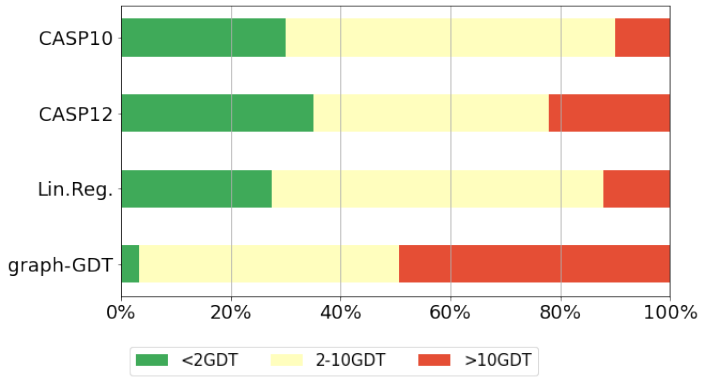


Figure 58: Percentages of the predictions with $\Delta GDT < 2$, $2 \leq \Delta GDT \leq 10$ and $\Delta GDT > 10$. The figures for CASP10 are averages of 12 best-performing models, and for CASP12 they are averages of 10 best-performing models.

accuracy, independent of other candidate structures. The consensus methods have generally outperformed the single-model methods, and this resulted in the development effort being concentrated on the consensus methods in the past [76]. More recently the single-model methods have received more attention and development effort [56], as the potential issues with the consensus methods are recognised. One issue, for example, is that the consensus methods may not be very useful in the environment outside the CASP-setup, where a large number of candidate structures may not be available for the input. Another potential issue, related to the first, is that the consensus methods may simply be taking advantage of the fact that many CASP models are now able to produce high-quality candidate structures, which are, naturally, similar to each other [96]. Our method is, by construction, unlikely to be improved to outperform the best accuracy estimation methods, as it ignores the geometric data in the candidate structures and only utilises the topological data. However, it may be able to

be combined with an existing method to improve its performance. Of course, it depends on a method to predict a target protein's hydrogen bonds by, for example, prediction of residue-residue contacts, where there has been a significant recent progress [84]. When a high-accuracy prediction of hydrogen bonds becomes possible, our method has the advantage that it could be combined with both a consensus method and a single-model method.

7 Local pattern and hydrogen bond rotation in proteins

In the previous two sections we have reviewed two projects for studying topology of proteins using fatgraph models. In the current section we will investigate the relation between topology of proteins in the form of graph structure consisting of the backbone and the hydrogen bonds, and geometry of proteins in the form of SO(3) rotations along hydrogen bonds (see section 4.2).

The space of possible protein structures appears vast, but when we look at the space of local structures, we see that the actual structures are restricted to certain regions in the space of all possible structures. One of the classical descriptor of protein local structures is a set of conformational angles φ and ψ at each C^α (see section 4.1). The well-known Ramachandran plot, where the actual conformational angles are plotted with φ in one axis and ψ in the other, shows clearly that the values are concentrated in relatively small regions [75]. Such concentration of the actual values is also observed, when we look at the spatial rotations between each pair of hydrogen bonded peptide planes [73]. Furthermore, it is found that these hydrogen bond rotations correspond well to the concrete secondary structures and other local structural motifs [73]. The question we wish to answer in this section is in some ways the opposite of the findings in [73]; given a protein's primary structure and its hydrogen bonds, can we predict SO(3) rotations along each of its hydrogen bonds? In other words, we wish to predict local geometry of a protein from its topology. In doing so, we focus on the local topological information around the bond in question, which we call H-bond local pattern (see section 7.1). It is a subgraph, centred around the hydrogen bond, of the graph consisting of the backbone and all hydrogen bonds in a protein. We discuss two different methodologies; one that tries to find an exact match for a given H-bond local pattern, and the other that finds a pattern, that best aligns with the given pattern.

7.1 H-bond local pattern

Our model of protein structure is as described in section 4.2. We draw the backbone horizontally with the N-terminus to the left. The half-edges representing the amino hydrogen are drawn below the backbone, and those representing the carboxyl oxygen are drawn above the backbone. Hydrogen bonds are represented by an edge between the corresponding half-edges. We show an example of such graph in figure 59. Often we think of the model as a sequence of three-atom segments N-C-O, and refer to the half-edge representing the amino hydrogen (below the backbone) as N-atom, and the carboxyl oxygen as O-atom. This construction also applies to a protein with multiple backbones, in which case the model is determined up to the permutation of backbones.

For a given hydrogen bond a, the *H-bond local pattern* or simply the *H-bond pattern* of *window size* w around a is the subgraph of the H-graph structure consisting of the set of backbone atoms whose distance along the backbone to one of the endpoints of a is no more than w atoms, together with all backbone and hydrogen bond edges between them. We call a the *central bond* of the pattern. The central bond determines the *signed length* of the bond, which is the distance, measured from the donor to the acceptor, between the central

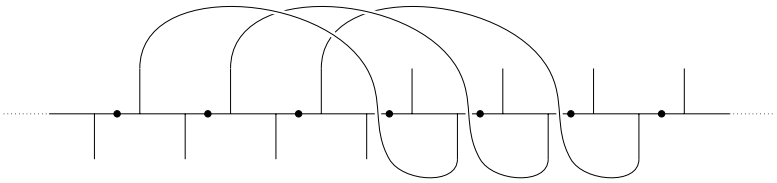


Figure 59: An example H-graph structure

bond's endpoints. Formally, if the donor of the bond is in the i'th peptide unit and the acceptor in the j'th peptide unit, the the signed distance d is defined to be $j - i$ if $i < j$ and $j - i - 1$ otherwise (figure 60). We will also record the hydrogen bonds whose two endpoints are no more than w atoms away from one of the endpoints of a along the backbone. An H-bond pattern may be expressed as a sequence of letters as follows (figure 61a);

```
bIXaIXIbXIaXII
```

Each pair of lowercase letters indicate a pair of atoms with a hydrogen bond between them. The letter "`a`" is given a special meaning as an indication of the central bonds. The remaining hydrogen bonds are ordered by the position of the donor atoms (starting from the N-terminus) and given letters "`b`, `c`, ... ". The uppercase letter `X` indicates a C^{α} atom, and the uppercase letter `I` indicates an "isolated" N or O atom (with no hydrogen bond attached). Hence we see that the above pattern corresponds to the structure shown in figure 59, with the second hydrogen bond as the central bond. If the distance between the two endpoints of the central bond is greater than $2w$, the H-bond pattern is not connected, which may be indicated by "`:`". So we may obtain a pattern such as (figure 61b);

```
IIXaIXb:IXIaXIb
```

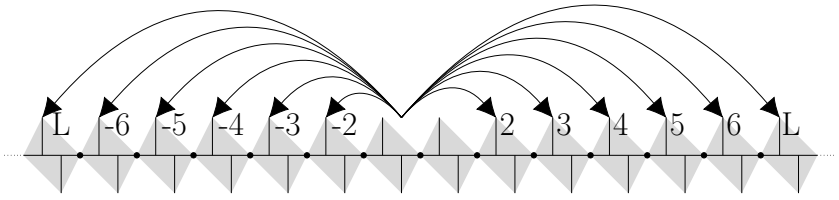
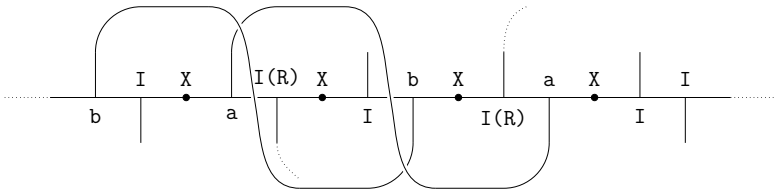


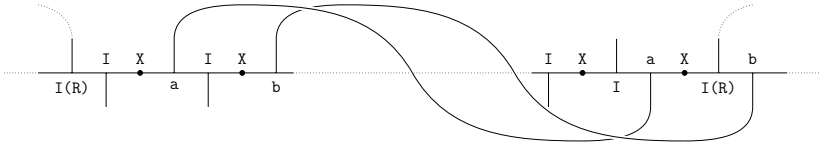
Figure 60: Signed length of hydrogen bonds. Peptide units are shaded grey. All bonds longer than ±6 are denoted by L.

This local pattern description can be enhanced further by encoding various extra pieces of information:

- A backbone atom within the window may have a hydrogen bond attached, whose other endpoint lies outside the window. In the basic pattern description, this would be indistinguishable from an unbonded backbone atom; using the letter `R` (for `R`emotely bonded) instead of `I` allows us to distinguish these cases. We may also restrict this information for atoms located



(a) Connected pattern. The pattern can be expressed as `bIXaIXIbXIaXII_-3`.



(b) Disconnected pattern. The pattern can be expressed as `IIXaIXb:IXIaXIb_L`.

Figure 61: Connected (figure 61a) and disconnected (figure 61b) H-bond local patterns of window size 3. The parts of backbone and hydrogen bonds, that do not participate in the pattern are shown by dotted lines. Note the remotely-bonded atoms may be replaced by the letter `R`, depending on the parameter specified (see text for details).

at most r from either endpoint of the central bond, for some $0 \leq r \leq w$. Atoms which are remotely bonded but lie further than r from the central bond are then treated as if they were isolated atoms.

- We may record whether a hydrogen bond is twisted or not. Formally, this is determined by whether the inner product between the normal vectors to the peptide planes is positive or negative. If the inner product is negative (that is, if the bond is twisted), we replace the lowercase letter with the corresponding letter from the other end of the alphabet (that is, `z` instead of `a`, `y` instead of `b` etc.). Similarly to the remote bonds, this information may be restricted to the atoms located at most t atoms from either endpoint of the central bond, for some $0 \leq t \leq w$.
- Beside the local pattern itself, we may record separately the residues at the four C^{α}'s closest to the central bond's endpoints. To reduce the number of possible bond descriptions and obtain reasonable clusters, we choose a grouping of the 20 residues into 1, 2, 3 or 4 groups according to their chemical properties (section 7.1), and simply record the 4-tuple of group identifiers.

The resulting pattern description may look as follows:

```
10XbRXcRXdRXzbXvcXIdXIvXRzXRRXRRXRRX_5_4LLLL
```
(58)

Here the number `10` indicates the window size, and the central bond is twisted; as shown by the use of letter `z` instead of `a`. There are several remotely-bonded atoms, indicated by the letter `R`. The last segment, `4LLLL`, indicates the number of groups used in the grouping of residues and the four group identifiers.

	No. of groups			
Amino acids	1	2	3	4
Leucine (L)				
Valine (V)				
Isoleucine (I)			L	L
Phenylalanine (F)				
Methionine (M)		L		
Alanine (A)				
Glycine (G)			A	A
Serine (S)				
Cysteine (C)				
Glutamic acid (E)				
Lysine (K)	X			
Arginine (R)				
Aspartic acid (D)				
Threonine (T)				E
Tyrosine (Y)		E	E	
Asparagine (N)				
Glutamine (Q)				
Histidine (H)				
Tryptophan (W)				
Proline (P)				P
N/A (X)		X	X	X

Table 15: Grouping scheme for amino acids and group labels.

The above description of H-bond pattern only relies on the H-graph structure of a protein. It is also possible to generate a local pattern which includes more information about a protein's structure. An *tertiary bond* is an object that identifies the closeness of various atoms in a tertiary structure, where no peptide or hydrogen bond is present. Note a tertiary bond is not necessarily a well-defined bond, but rather an indication of presence of some interaction, inferred from the tertiary structure. Suppose we have a protein's tertiary bond information along with its secondary structure. Then we may include in a pattern all atoms that are no more than w away from either end of the central bond, measured along the backbone and the tertiary bonds. H-bond patterns produced in this way may better reflect the local structure around the central bond. A pattern produced in this way can no longer be presented in the form (58), but may be presented as a graph.

7.2 Rotation prediction by H-bond pattern matching

7.2.1 Method

Our analysis is done in two phases; training and prediction. We start with the description of the training phase. The training dataset consists of a collection of proteins whose geometric structures, in particular the rotations along hydrogen bonds, are known. We then perform the local pattern identification on each hydrogen bond using various combinations of the parameters;

- Window size
- Whether to indicate remote bonds, and how far from the central bond
- Whether to indicate twisted bonds, and how far from the central bond
- The number of residue classes

Thus we obtain, for each hydrogen bond in the training dataset, a number of H-bond patterns, one for each parameter combination. These are then grouped, so that each set contains only the identical H-bond patterns. We may, at this point, discard the patterns with the number of occurrences fewer than some predetermined threshold. This will increase our chance of obtaining a reasonable cluster later on in the analysis. For each of the remaining H-bond patterns, we collect the associated SO(3) rotations for each bond, which is then fed into a clustering algorithm. Each of these clustering runs is then evaluated and assigned a score, which reflects the extent to which all observed bonds with the same H-bond pattern fall into a single well-defined cluster.

A description of the clustering algorithm can be found in [73] (Method section). We note that the algorithm uses a discretised rotational space; it divides the cube $(-\pi, \pi)^3$ into $81 \times 81 \times 81$ small boxes, and finds a mode box for each well-defined cluster. Each box can belong to at most one cluster, even when the algorithm finds several clusters.

The score s for each clustering run is determined by the following formula;

$$s = \begin{cases} \pi - m & \text{if there is only one cluster} \\ -1 & \text{if there are more than one cluster,} \end{cases} \tag{59}$$

where m is the mean distance between all boxes in the cluster and the mode box, which is by definition bounded by π. In this way we associate a score for each bond description. The result is a table where each row contains an H-bond pattern, a rotation value (which is the centre of the mode box of the largest cluster) and a score. This data is then used to predict rotation of a given hydrogen bond from a topological model of the protein.

Prediction is done using the same procedure as the training, but applied on a protein with unknown geometric structure. Suppose we have a protein whose backbone sequence and the set of hydrogen bonds are known. In other words, the protein's H-graph structure is assumed to be known, but not its tertiary structure. For each hydrogen bond in the protein, we obtain H-bond patterns using the same sets of parameter combinations as used in the training stage. Each resulting H-bond pattern is looked up in the table of clustering results. If a match is found, we obtain an estimate for the bond's rotation, which is the centre of the largest cluster, along with a score for that estimate, which is the score associated to the cluster. Our final prediction for the rotation associated to the hydrogen bond is the estimate with the highest associated score. If two estimates have the same score, the result with more detailed H-bond pattern is used for the prediction.

7.2.2 Results

The dataset used for the analysis was a collection of 8182 proteins of known geometric structures taken from the protein data bank [20], containing approximately 1.16 million hydrogen bonds. The dataset was processed in the way

Range of d		Run 1	Run 2	Run 3	Total
$0 \leq d < 0.2664$	(0.1%)	60.41	58.55	59.44	59.47
$0.2664 \leq d < 0.4567$	(0.5%)	80.30	78.89	79.55	79.58
$0.4567 \leq d < 0.5766$	(1.0%)	86.61	85.75	86.08	86.15
$0.5766 \leq d < 0.7862$	(2.5%)	92.45	92.42	92.26	92.38
$0.7862 \leq d < 0.9968$	(5.0%)	95.32	95.13	95.29	95.24
$0.9968 \leq d < 1.2689$	(10.0%)	97.23	96.82	97.14	97.06
$1.2689 \leq d < 1.7663$	(25.0%)	98.73	98.56	98.71	98.67
$1.7663 \leq d < 2.3099$	(50.0%)	99.72	99.62	99.65	99.66
$2.3099 \leq d < 2.7437$	(75.0%)	99.89	99.91	99.87	99.89
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00	100.00	100.00
Number of bonds		25612	25623	23998	75233

Table 16: Accumulative % of hydrogen bonds whose predicted rotation values lie within the specified distance from the true rotation values. The numbers in parentheses show the volume of the ball whose radius is the upper limit of the range as a proportion of the volume of entire SO(3), w.r.t. the invariant metric on SO(3).

described in section 5.6. A subset of 200 proteins was randomly selected from the collection, and the training procedure was performed on the remaining 7982 proteins. The H-bond pattern identification was performed with the following combinations of parameters;

- Window size: 0 (only the central bond), $1, 2, \ldots, 10$
- Remote bonds: 0 (do not indicate remote bond), $1, 2, \ldots,$ window size
- Twisted bonds: none (-1), only the central bond (0), window size
- Number of residue classes: 1, 2, 3, 4

This resulted in 792 parameter combinations. For each of these parameter combinations, the H-bond patterns with less than 30 occurrences were discarded. For each of the remaining H-bond patterns with the associated SO(3) rotations, we performed the clustering analysis (see [73] for the description of clustering algorithm), and computed a score for each of them.

Next, we applied the prediction procedure outlined above to the remaining 200 proteins. We assessed the quality of each prediction by measuring the distance between the predicted and the actual rotation. The analysis was repeated three times with different choices of 200 proteins to be used for prediction. In the following, unless otherwise specified, the accuracy figures quoted are averages over three runs. The results are shown in table 16.

We see in 59.47% of all cases, the predicted rotation lies within a ball comprising just 0.1% of the total volume of SO(3) centred at the true rotation, and in 86.15% of all cases, the prediction was within a ball corresponding to 1% of the volume of SO(3). Since the size of PDB is continuously increasing, we updated our dataset in 2017 to obtain an expanded dataset with 13115 proteins, containing approximately 1.89 million hydrogen bonds. The results from this expanded dataset were very similar, with 85.99% of the prediction lying inside a

Range of d		Dataset 1	Dataset 2
$0 \leq d < 0.2664$	(0.1%)	59.47	59.26
$0.2664 \leq d < 0.4567$	(0.5%)	79.58	79.52
$0.4567 \leq d < 0.5766$	(1.0%)	86.15	85.99
$0.5766 \leq d < 0.7862$	(2.5%)	92.38	92.29
$0.7862 \leq d < 0.9968$	(5.0%)	95.24	95.17
$0.9968 \leq d < 1.2689$	(10.0%)	97.06	97.16
$1.2689 \leq d < 1.7663$	(25.0%)	98.67	98.69
$1.7663 \leq d < 2.3099$	(50.0%)	99.66	99.66
$2.3099 \leq d < 2.7437$	(75.0%)	99.89	99.85
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00

Table 17: Accumulative % of hydrogen bonds whose predicted rotation values lie within the specified distance from the true rotation values, comparison between the old (Dataset 1) and the new (Dataset 2) datasets

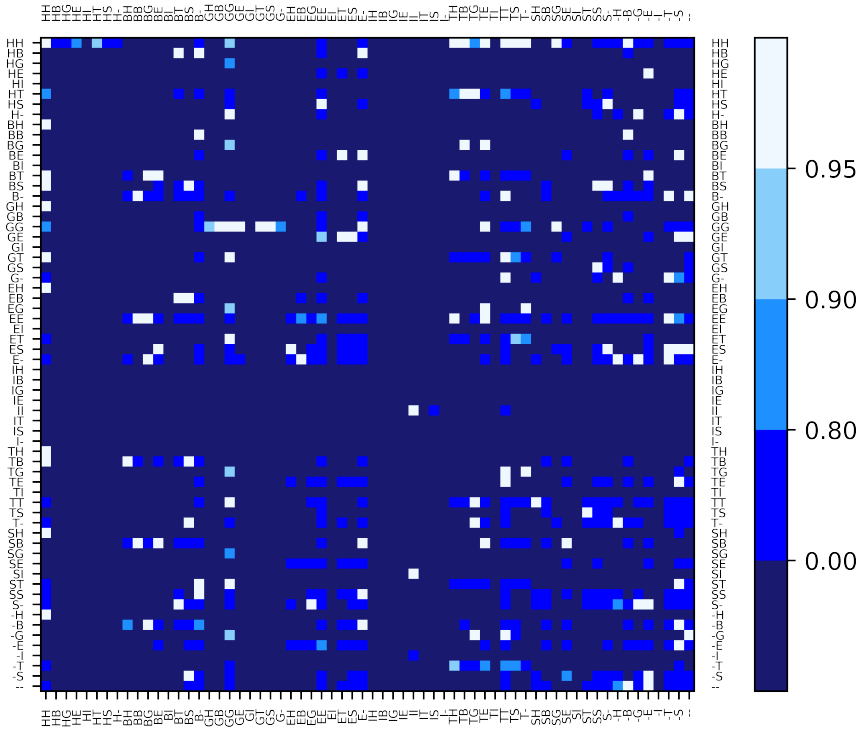
DSSP class	DSSP symbol	Local structure
α-helix	H	Helix
3_{10}-helix	G	
π-helix	I	
Strand	E	Sheet
Isolated β-bridge residue	B	Coil
Turn	T	
Bend	S	
Unclassified	-	Unclassified

Table 18: DSSP classes and corresponding local structure patterns.

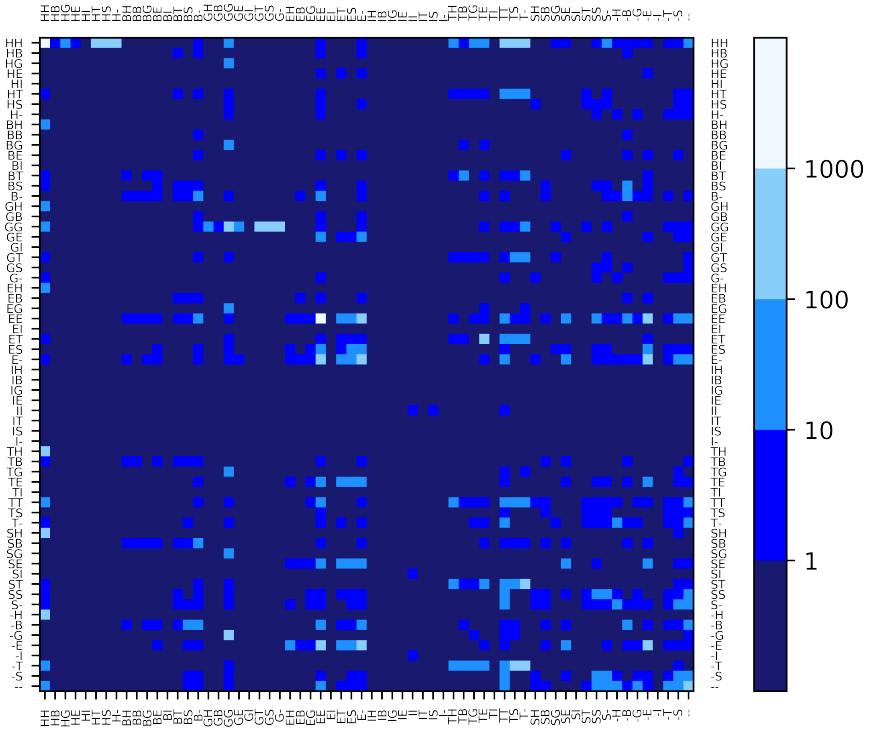
ball corresponding to 1% of the volume of SO(3) (table 17). We have therefore chosen to perform the following analysis using the smaller dataset due to the extra time required to analyse the larger dataset. To analyse the prediction results further, we have looked at the DSSP classes [51] of four residues around each hydrogen bond. DSSP assigns seven secondary structure classes (plus "unclassified") to each residue (table 18). The four residues around a hydrogen bond are chosen and recorded in the following order;

- The residue preceding the N-donor amino acid residue.
- The N-donor amino acid residue.
- The O-acceptor amino acid residue.
- The residue following the O-acceptor amino acid residue.

We analysed the frequencies of four-tuples of DSSP classes and the associated prediction accuracies (figure 62). We can observe the frequencies concentrated on a few classes, which is also evident if we look at the ten most frequent DSSP class combinations (table 19). We also observe that the residue class combination for the sheet structure ("EEEE") has the high frequency and relatively low accuracy.



(a) Accuracy of predictions by the DSSP classes of four residues around the hydrogen bond; % of predictions lying within 1% SO(3) volume.



(b) Frequency by the DSSP classes of four residues around the hydrogen bond.

Figure 62: Accuracy and frequency of predictions by DSSP classes. The horizontal and vertical axes show the donor-side and the acceptor-side residues, respectively.

Residues around H-bond	Frequency	Accuracy
HHHH	9296	99.83%
EEEE	5604	79.59%
HH-H	780	99.36%
HTHH	694	97.84%
T–T	405	80.74%
TTHH	390	96.15%
T-HH	339	88.79%
GG-G	290	93.45%
TS-T	281	87.54%
H-HH	263	91.25%

Table 19: 10 most frequent combinations of DSSP classes around hydrogen bonds, together with the proportions of predictions, which lie inside a ball centred at the true rotation having a volume corresponding to 1% of the total volume of SO(3).

We then analysed each prediction and looked at the parameters used to produce it. By looking at the distance between predicted and true rotations (Δ) and the mean distance to cluster mode (m) for each prediction, we found a group of predictions made using small window sizes, with many of them having large Δ values (figure C.1). This could happen, for example, if the H-bond pattern matched with a smaller window size has the associated cluster with a well-defined "peak" (thus having a low m value and high score), while the cluster for a match with a larger window size has a lower "peak" (and a high m value). To encourage the use of larger pattern for prediction, we modified the score function (59) to penalise the use of smaller patterns. The new score function is given by

$$s = \begin{cases} \pi - m - \exp(3 - w) & \text{if there is only one cluster} \\ -1 & \text{if there are more than one cluster,} \end{cases} \tag{60}$$

where m is the mean distance to the cluster mode, and $m = \max\{3, \text{window size}\}$. Using the new score function, we achieved the prediction accuracy of 89.05% inside 1% SO(3) volume (table 20). Analyses of the other parameters (Remote bonds, Twisted bonds, and Residue groups) did not show any similar anomalies, and applying similar modifications to the score function to prioritise matches with more detailed patterns did not result in an improvement.

We have also run the analysis using the local H-bond patterns generated including the tertiary bond information, where a tertiary bond counts as one backbone bond in computing window size, and we do not consider atoms more than one tertiary bond away from the central bond. In other words, an atom is only included in the H-bond pattern if the distance (along the backbone and the tertiary bond edges) from it to either end of the central bond is less than or equal to the window size, and that no more than one tertiary bond is traversed in computing the distance. This criteria was applied to limit the pattern to the atoms most likely to exert some influence on the central H-bond. We found that 82.01% of all predictions lay within 1% SO(3)-volume of true values (table 20).

Range of d		Reference	New score	Tertiary
$0 \leq d < 0.2664$	(0.1%)	59.47	64.75	53.16
$0.2664 \leq d < 0.4567$	(0.5%)	79.58	83.47	74.51
$0.4567 \leq d < 0.5766$	(1.0%)	86.15	89.05	82.01
$0.5766 \leq d < 0.7862$	(2.5%)	92.38	94.02	89.70
$0.7862 \leq d < 0.9968$	(5.0%)	95.24	96.24	93.60
$0.9968 \leq d < 1.2689$	(10.0%)	97.06	97.64	96.16
$1.2689 \leq d < 1.7663$	(25.0%)	98.67	98.99	98.16
$1.7663 \leq d < 2.3099$	(50.0%)	99.66	99.77	99.69
$2.3099 \leq d < 2.7437$	(75.0%)	99.89	99.93	99.95
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00	100.00

Table 20: Accumulative % of hydrogen bonds whose predicted rotation values lie within the specified distance from the true rotation values, for reference data (table 16), with the new score function (60), and with tertiary bonds.

7.3 Rotation prediction by H-bond pattern alignment

In this section we describe another analysis utilising H-bond local patterns. Instead of trying to find an exact match in the list of H-bond patterns generated from known structures, we find the H-bond pattern that best aligns with the H-bond pattern, whose rotation is to be predicted. To do this, we translate a given H-bond pattern to a sequence indicating the presence or absence of hydrogen bond together with its length, if present. Alignment score between two such sequences are computed using Needleman-Wunsch algorithm [66].

7.3.1 Method

We start again with the training dataset with known geometric structures. An H-bond pattern is computed for each hydrogen bond in the training dataset, in the same manner as in the previous section. It is however not necessary to use multiple parameter combinations, since in the current analysis an exact match is not required. Each H-bond pattern is then translated to a sequence, which we call an H-bond sequence, as follows (see figure 63).

1. Recall that each H-bond pattern corresponds to a repeated sequence of N-C^{α}-O atoms, representing an amino acid, in our protein model. At either end of the pattern we may have only a part of this three-atoms sequence; for example it may start with C^{α} or O atom. Given an H-bond pattern, we split it into three-atom segments corresponding to amino acids (we may end up with segments at either end of the pattern which consist of fewer than three atoms). In the following procedure we only consider the atoms in the H-bond pattern.

2. For each of the resulting segments, check whether the N atom is a donor. If so, we assign a symbol encoding the signed length followed by the twistedness of the bond ("+" for twisted, and "-" for not twisted). E.g. "+4-" for a bond of length +4, which is not twisted ("-").

3. For each of the remaining segments, check whether the O atom is an acceptor. If so, we assign a symbol "A" to the segment.

4. The remaining segments are assigned a symbol "U" for unbonded.

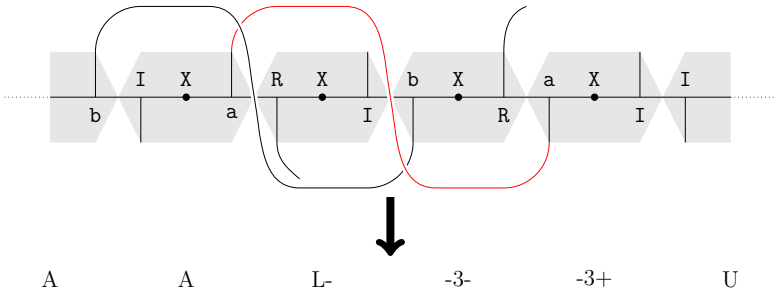


Figure 63: Translation of H-bond pattern to H-bond sequence. Amino acid residues are shaded grey. A twisted bond is shown in red.

Using the above translation method, we can compute the list of H-bond sequence and associated rotation value from our training dataset. The same procedure is applied to the proteins whose H-bond rotations are to be predicted. For an H-bond a in the prediction dataset, let $s(a)$ be the corresponding H-bond sequence. Our prediction for the rotation along a is the rotation value associated to the H-bond sequence which best aligns with $s(a)$.

7.3.2 Results

The dataset used was the same as in section 7.2.2, with 200 proteins selected for prediction. The parameter combination used to compute H-bond patterns were; window size = 10, remote bonds = 10, twisted bonds = 10, residue class = 1. Note the parameter residue class does not have any effect on the resulting H-bond sequence. The substitution matrix used for the Needleman-Wunsch algorithm was computed as follows. Let

$$\begin{aligned} S_1 &= \{-6-, -5-, \ldots, +5-, +6-, -6+, -5+, \ldots, +5+, +6+\} \\ S_2 &= \{L-, L+\} \\ S &= S_1 \cup S_2 \\ K &= S \cup \{\mathrm{U}, \mathrm{A}\}. \end{aligned}$$

Define $l : S \to \mathbb{Z} \cup \{L\}$ to be the function that returns the length part of $x \in S$, i.e. it removes the last character in x. Define also $t : S \to \{+, -\}$ to be the function that returns the twistedness of $x \in S$. Construct the substitution matrix M with entries M_{k_1,k_2}, $k_1, k_2 \in K$ by the pseudocode shown in algorithm 5. The gap score was set to -1.

The results of the analysis is shown in table 21, under the column titled "Linear". We could only achieve accuracy of 72.91% inside 1% SO(3) volume.

To investigate the effect of the difference between the lengths and twistedness of bonds when a replacement occurs, we have run the analysis with modified substitution matrices. In the first the penalty for when $k_1, k_2 \in S_1$ ("short to

Algorithm 5 Pseudocode for the construction of substitution matrix M.

1: **for** $k_1 \in K$ **do**
2: **for** $k_2 \in K$ **do**
3: **if** $k_1 == k_2$ **then**
4: $s = 1$
5: **else if** $\{k_1, k_2\} \cap (K \setminus S) \neq \emptyset$ **then**
6: $s = -1$
7: **else if** $l(k_1) == l(k_2)$ **then**
8: $s = 0$
9: **else if** $\{k_1, k_2\} \cap S_2 \neq \emptyset$ **then**
10: $s = -0.75$
11: **else**
12: $s = -\big|l(k_1) - l(k_2)\big| /20$
13: **end if**
14: **if** $\{k_1, k_2\} \subset S$ and $t(k_1) \neq t(k_2)$ **then**
15: $s = s - 0.1$
16: **end if**
17: $M(k_1, k_2) = s$
18: **end for**
19: **end for**

short" substitution) was made exponential instead of linear, by replacing line 12 with

$$d = -\big|l(k_1) - l(k_2)\big|$$
$$s = -0.6\left((\exp d - 1)/(\exp 12 - 1)\right).$$

In the second the penalty was made logarithmic by replacing line 12 with

$$d = -\big|l(k_1) - l(k_2)\big|$$
$$s = -0.6\left(\log(d+1)/\log(12+1)\right).$$

Finally, the effect of twistedness was tested by replacing line 15 by

$$s = s - 0.8.$$

The results for the three substitution matrices are shown in table 21.

We also investigated the effect of different gap scores by using the simplest substitution matrix; 1 along the diagonal (match) and -1 elsewhere (mismatch), with different gap scores. Perhaps surprisingly, the prediction accuracy using this simple substitution matrix was very similar to the result obtained by using large twistedness penalty (table 22). There was a reduction in accuracy when the gap score was set to 0, and a relatively large improvement when it was set to -5.

7.4 Discussion

In the first method, where the rotation prediction was done by finding an exact match for a given H-bond pattern, we were able to achieve close to 90% of our predictions lying inside 1% SO(3) volume of the true rotations. We believe there are potentials for further improvement, since an analysis of the clustering results show that if we could choose the "best" clustering result, i.e. the clustering result that lies closest to the true value, it will result in over 97% of "predictions" inside 1% SO(3)-volume. So it is possible that a better score function than (59) may

Range of d		Linear	Exp	Log	Twisted
$0 \leq d < 0.2664$	(0.1%)	49.56	49.57	49.64	49.74
$0.2664 \leq d < 0.4567$	(0.5%)	66.49	66.47	66.55	66.91
$0.4567 \leq d < 0.5766$	(1.0%)	72.91	72.91	72.98	73.55
$0.5766 \leq d < 0.7862$	(2.5%)	80.65	80.66	80.72	81.48
$0.7862 \leq d < 0.9968$	(5.0%)	85.33	85.34	85.39	86.22
$0.9968 \leq d < 1.2689$	(10.0%)	89.17	89.16	89.21	90.02
$1.2689 \leq d < 1.7663$	(25.0%)	94.49	94.47	94.50	95.11
$1.7663 \leq d < 2.3099$	(50.0%)	97.48	97.47	97.50	98.05
$2.3099 \leq d < 2.7437$	(75.0%)	98.86	98.85	98.86	99.24
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00	100.00	100.00

Table 21: Accumulative % of hydrogen bonds whose predicted rotation values lies within the specified distance from the true rotation values, for different substitution matrices. The numbers in parentheses show the volume of the ball whose radius is the upper limit of the range as a proportion of the volume of entire SO(3), w.r.t. the invariant metric.

give the necessary improvement. It is also clear from the results of the DSSP class analysis, that we need to improve our predictions in the sheet structure, if we are to reach our goal. Our hypothesis is that due to the structural flexibility of sheets compared to helices, the bonds in sheets are affected by the nearby atoms (which are not necessarily near the bond in the secondary structure) to a greater extent than the bonds in helices. We have attempted, unsuccessfully so far, to take this into account by using tertiary bond information in our H-bond pattern generation. A better understanding of β-sheet structures and their topology may be needed to improve prediction of hydrogen bond rotations inside β-sheets.

The prediction accuracy was worse in the second method, where H-bond pattern alignment score was used to find the best match. Changing the substitution matrices to simulate non-linear penalties did not result in any significant change in prediction accuracy. A minor improvement was observed when the penalty for the twistedness was increased (table 21). This may indicate the fact that the twistedness of a hydrogen bond is directly related to its associated rotation value; if a bond is twisted and another is not twisted, it is unlikely that the two have similar rotation values. More surprisingly, an improvement similar to using large twistedness penalty was observed when using the "simple" substitution matrix where a match is given the score of 1 and a mismatch -1. A further, even larger improvement was observed by using the gap score of -5. As the window size is constant for the training and test datasets, larger gap score has the effect of making the algorithm more like the one that simply counts the match/mismatch between the two H-bond sequences. It is however not immediately clear why this may increase the prediction accuracy compared to allowing gaps, or making penalties dependent on the change in the bond lengths. It should be noted, that even though the Needleman-Wunsch algorithm itself does not place any restriction on its use, in the study of protein structures it is typically used for aligning segments of primary sequences. Accordingly, there are substitution matrices which are generally accepted in the field and tested

Range of d		Gap=-1	Gap=-5	Gap=0
$0 \leq d < 0.2664$	(0.1%)	50.01	51.31	47.19
$0.2664 \leq d < 0.4567$	(0.5%)	67.12	69.55	62.97
$0.4567 \leq d < 0.5766$	(1.0%)	73.71	76.67	69.46
$0.5766 \leq d < 0.7862$	(2.5%)	81.80	84.94	77.41
$0.7862 \leq d < 0.9968$	(5.0%)	86.60	89.74	82.39
$0.9968 \leq d < 1.2689$	(10.0%)	90.43	93.08	86.43
$1.2689 \leq d < 1.7663$	(25.0%)	95.47	96.24	93.21
$1.7663 \leq d < 2.3099$	(50.0%)	98.27	98.31	96.64
$2.3099 \leq d < 2.7437$	(75.0%)	99.29	99.22	98.48
$2.7437 \leq d < 3.1416$	(100%)	100.00	100.00	100.00

Table 22: Accumulative % of hydrogen bonds whose predicted rotation values lie within the specified distance from the true rotation values, for different gap scores. The numbers in parentheses show the volume of the ball whose radius is the upper limit of the range as a proportion of the volume of entire SO(3), w.r.t. the invariant metric.

in various applications [63]. The sequences considered in this study have not been studied previously, and the substitution matrix was developed specifically for this study. The fact that we were still able to achieve a prediction accuracy of more than 75% lying inside 1% SO(3) volume of the true rotations indicates that the sequence consisting of bond lengths contains a certain amount of information about H-bond rotations. It is also possible that a further development of the substitution matrix may improve the prediction accuracy.

8 Future perspectives

In the three projects discussed in this thesis, we have investigated how our fatgraph models of proteins can contribute to our understanding of protein structures in relation to the protein folding problem. We started with focusing on the topology of β-sheets, an important protein sub-structure. We have shown our filter based on the topology of β-sheets is able to substantially reduce the number of possible candidate structures while retaining the best candidates. Our method, when combined with another method that does not utilise the topology of β-sheets, was able to improve the prediction accuracy. We then looked at the topology of entire proteins, in an experiment inspired by CASP Model Quality Assessment. With the extra information about hydrogen bonds which is not available in CASP experiment, we showed that the topology of proteins can be used to predict the geometric structure closest to the target structure to an accuracy level comparable to the best performing models in CASP10 and CASP12. Indeed, it seems clear that topology of proteins is important in their foldings, after seeing how the fatgraph model produced positive results in the study of RNA structures, and the difference between topology of simulated protein metastructure data and the data generated from PDB. We do not know at this stage, whether the current method can be improved to achieve better results, or that we should be looking at a different experiment to apply our ideas. Lastly we investigated the extent to which proteins' local topology determines their local geometry. We were able to determine the local geometry, expressed as the rotation along each hydrogen bond, to a high accuracy level, even though our results were unlikely to be able to match the recent achievements by AlphaFold2 [49], although a direct comparison is not possible as we do not produce the final prediction of the folded structures. Nonetheless, our results points to the important link between topology and geometry of proteins.

The three projects are based on the theoretical foundations, which we also discussed in this thesis. We have used the fatgraph model of proteins introduced in [72] to prove the recursion relation for the number of protein diagrams, both combinatorially and by the matrix model, which we also constructed in this thesis. A possiblity for the future development of the project is to investigate whether the 2- (and 3-) matrix models we constructed here satisfy topological recursion. 2-matrix models have been extensively studied and there are number of literatures by Chekhov, Eynard, Orantin and others [28, 42, 39, 35, 41, 40, 43, 24]. But as far as we are aware, the potentials studied for these models have been of the form

$$AB + V_1(A) + V_2(B).$$

So it seems a further invesigation into the protein matrix model could lead to a new development in relation to topological recursion.

When we consider more practical appication, there are further challenges if the fatgraph model of proteins is to be truly useful for the study of protein structures and foldings. The first of which is the chemical diversity of amino acid residues. Our protein model, in its most basic form, only models the topological structure of a protein, and does not consider the residues. However, the residues, in particular their size and their chemical characteristics, are clearly important in determining the native structure of a protein, and it will be necessary to consider them in the future development of our proposed programme. This is

also true in the case of RNA structures, but in RNA there are only 4 types of nucleotides, and we know that the formation of hydrogen bonds is limited to between adenine and uracil, guanine and cytosine, and uracil and guanine. Proteins consist of 20 different residues, and there is no deterministic rule for hydrogen bond formation similar to the Watson-Crick rules. As the number of theoretically possible sequence increases exponentially with their lengths, it is not difficult to imagine the computation will be too onerous much earlier with proteins than with RNA. This issue could be addressed by considering each residue's propensity to form a hydrogen bond, or to participate in certain secondary structure [22, 65, 97]. Then different weights can be applied to the protein graph structures to account for the different primary sequences.

The second challenge is related to the relative structural rigidity of proteins. As described in section 4.1, there is an intrinsic rigidity in the protein backbone that comes from the structure of peptide units. This rigidity will make some conformations energetically unfavourable, and in some cases prevent certain conformations. This obstruction manifests itself in some of the empty regions in the Ramachandran plot [75]. The third challenge is the propensity of proteins to form certain recurring structures, such as α-helices and β-sheets [5]. This implies that the formation of hydrogen bonds in proteins is not random, but to a large extent linked to the formation of these recurring structures. This, in the context of our approach, means that the formation of hydrogen bonds that participate in these structures should be favoured over those that result in less common structures, while still being compatible with desirable overall shapes for the protein under consideration. These two issues can be addressed to an extent, if we can identify the grammar for construction of protein graph structures, similar to [7]. With the decomposition grammar, a (pseudo-)energy functional can be constructed, which assigns appropriate energy cost for each basic structures and for each "move" needed to construct a given protein graph structure. The energy cost can be computed from the relative frequencies of the structure in question in the existing database of protein structures, such as PDB [20].

Despite these challenges, there is a strong motivation for pursuing the use of the protein fatgraph model in the folding problem, and more generally, in the study of protein structures. Apart from the mathematical interest in finding a novel application of fatgraph and matrix model theories, it provides a common language for describing and studying various biological macromolecules. For the time being the application is limited to RNA and proteins, but there are possibilities for extending fatgraph model to, for example, study of polysaccharides [69]. Having a common language for studying RNA and protein structures is particularly useful, as a class of proteins (called RNA-binding proteins) are thought to participate in the regulation of RNAs during and after transcription [60, 52]. The importance of these protein-RNA interactions are becoming clearer with the advance in the methods for analysing them [53, 91]. A common modelling framework for RNA and protein could be an ideal tool for studying these interactions. Our hope is that the fatgraph model of proteins will provide a foundation for the common framework, and contribute to the fuller understanding of this fascinating field.

References

[1] 't Hooft, Gerard. "A planar diagram theory for strong interactions". *Nuclear Physics B* 72 (1974), pp. 461–473.

[2] Alderson, T Reid, Lee, Jung Ho, Charlier, Cyril, Ying, Jinfa, and Bax, Ad. "Propensity for cis-proline formation in unfolded proteins". *Chembiochem: a European journal of chemical biology* 19.1 (2018), p. 37.

[3] Alexeev, Nikita, Andersen, Jørgen Ellegaard, Penner, Robert C., and Zograf, Peter. "Enumeration of chord diagrams on many intervals and their non-orientable analogs". *Advances in Mathematics* 289 (Feb. 2016), pp. 1056–1081. ISSN: 0001-8708. DOI: doi:10.1016/j.aim.2015.11.032.

[4] Allen, Frances, Almasi, G, Andreoni, Wanda, Beece, D, Berne, Bruce J., Bright, A, Brunheroto, Jose, Cascaval, Calin, Castanos, J, Coteus, Paul, et al. "Blue Gene: A vision for protein science using a petaflop supercomputer". *IBM systems journal* 40.2 (2001), pp. 310–327.

[5] Almeida, Paulo. *Proteins: concepts in biochemistry*. Garland Science, 2016.

[6] Altman, Sidney. "Enzymatic cleavage of RNA by RNA". *Bioscience reports* 10.4 (1990), pp. 317–337.

[7] Andersen, J. E., Huang, F. W. D., Penner, R. C., and Reidys, C. M. "Topology of RNA-RNA interaction structures". *Journal of Computational Biology* 19.7 (2012), pp. 928–943.

[8] Andersen, J. E., Penner, R. C., Reidys, C. M., and Waterman, M. S. "Topological classification and enumeration of RNA structures by genus". *Mathematical Biology* 67 (2013), pp. 1261–1278.

[9] Andersen, Jørgen E, Chekhov, Leonid O., Penner, Robert C, Reidys, Christian, and Sułkowski, Piotr. "Enumeration of RNA complexes via random matrix theory". *Biochemical Society. Transactions* 41.2 (2013), pp. 652–655. ISSN: 0300-5127. DOI: 10.1042/BST20120270.

[10] Andersen, Jørgen Ellegaard, Borot, Gaëtan, Chekhov, Leonid O., and Orantin, Nicolas. *The ABCD of topological recursion*. WorkingPaper. arXiv.org, June 2017.

[11] Andersen, Jørgen Ellegaard, Borot, Gaëtan, and Orantin, Nicolas. "Modular functors, cohomological field theories and topological recursion". English. In: *Proceedings of Symposia in Pure Mathematics*. Ed. by Chiu-Chu Melissa Liu and Motohico Mulase. Vol. 100. Proceedings of Symposia in Pure Mathematics. United States: American Mathematical Society, 2018, pp. 1–58. ISBN: 978-1-4704-3541-7.

[12] Andersen, Jørgen Ellegaard, Chekhov, Leonid O., Penner, Robert, Reidys, Christian M., and Sułkowski, Piotr. "Topological recursion for chord diagrams, RNA complexes, and cells in moduli spaces". *Nuclear Physics, Section B* 866.3 (2013), pp. 414–443. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2012.09.012.

[13] Andersen, Jørgen Ellegaard, Fuji, Hiroyuki, Manabe, Masahide, Penner, Robert C., and Sułkowski, Piotr. "Enumeration of chord diagrams via topological recursion and quantum curve techniques". *Travaux Mathematiques, University of Luxembourg* 25 (Mar. 2017), pp. 285–323.

[14] Andersen, Jørgen Ellegaard, Fuji, Hiroyuki, Manabe, Masahide, Penner, Robert C., and Sułkowski, Piotr. "Partial Chord diagrams and Matrix models". English. *Travaux Mathematiques, University of Luxembourg* 25 (Mar. 2017), pp. 233–283.

[15] Andersen, Jørgen Ellegaard, Fuji, Hiroyuki, Penner, Robert C., and Reidys, Christian. *The boundary length and point spectrum enumeration of partial chord diagrams using cut and join recursion.* WorkingPaper. arXiv.org, Dec. 2016.

[16] Anfinsen, Christian B. "Studies on the principles that govern the folding of protein chains". *Les Prix Nobel en 1972* (1971), pp. 103–119.

[17] Aramayona, Javier. "Hyperbolic structures on surfaces". In: *Geometry, topology and dynamics of character varieties.* World Scientific, 2012, pp. 65–94.

[18] Aydin, Zafer, Altunbasak, Yucel, and Erdogan, Hakan. "Bayesian Models and Algorithms for Protein β-Sheet Prediction". *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 8.2 (2011), pp. 395–409.

[19] Baker, E.N. and Hubbard, R.E. "Hydrogen bonding in globular proteins". *Progress in Biophysics and Molecular Biology* 44.2 (1984), pp. 97–179. ISSN: 0079-6107. DOI: https://doi.org/10.1016/0079-6107(84)90007-5. URL: http://www.sciencedirect.com/science/article/pii/0079610784900075.

[20] Berman, Helen M., Westbrook, John, Feng, Zukang, Gilliland, Gary, Bhat, T. N., Weissig, Helge, Shindyalov, Ilya N., and Bourne, Philip E. "The Protein Data Bank". *Nucleic Acids Research* 28.1 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235. URL: http://dx.doi.org/10.1093/nar/28.1.235.

[21] Bessis, D., Itzykson, C., and Zuber, J. B. "Quantum Field Theory Techniques in Graphical Enumeration". *Advances in Applied Mathematics* 1 (1980), pp. 109–157.

[22] Blaber, Michael, Zhang, Xue J, and Matthews, Brian W. "Structural basis of amino acid alpha helix propensity". *Science* 260.5114 (1993), pp. 1637–1640.

[23] Bon, Michael, Vernizzi, Graziano, Orland, Henri, and Zee, A. "Topological classification of RNA structures". *Journal of molecular biology* 379.4 (2008), pp. 900–911.

[24] Borot, Gaëtan, Eynard, Bertrand, and Orantin, Nicolas. "Abstract loop equations, topological recursion and new applications". *Communications in Number Theory and Physics* 9.1 (2015), pp. 51–187.

[25] Brézin, Edouard, Itzykson, Claude, Parisi, Giorgio, and Zuber, Jean-Bernard. "Planar diagrams". *Communications in Mathematical Physics* 59 (), pp. 35–51.

[26] Cech, TR. "Self-splicing and enzymatic activity of an intervening sequence RNA from Tetrahymena." *Bioscience reports* 10.3 (1990), p. 239.

[27] Chekhov, Leonid and Eynard, Bertrand. "Matrix eigenvalue model: Feynman graph technique for all genera". *Journal of High Energy Physics* 2006.12 (2006), p. 026.

[28] Chekhov, Leonid, Eynard, Bertrand, and Orantin, Nicolas. "Free energy topological expansion for the 2-matrix model". *Journal of High Energy Physics* 2006.12 (2006), p. 053.

[29] Cheng, Jianlin and Baldi, Pierre. "Three-stage prediction of protein ß-sheets by neural networks, alignments and graph algorithms". *Bioinformatics* 21.suppl-1 (June 2005), pp. i75–i84. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti1004. URL: https://doi.org/10.1093/bioinformatics/bti1004.

[30] Chothia, Cyrus, Hubbard, Tim, Brenner, Steven, Barns, Hugh, and Murzin, Alexey. "Protein folds in the all-β and all-α classes". *Annual review of biophysics and biomolecular structure* 26.1 (1997), pp. 597–627.

[31] Dehghani, Toktam, Naghibzadeh, Mahmoud, and Sadri, Javad. "Enhancement of Protein β-sheet Topology Prediction using Maximum Weight Disjoint Path Cover". *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.6 (2018), pp. 1936–1947.

[32] Dill, Ken A and MacCallum, Justin L. "The protein-folding problem, 50 years on". *science* 338.6110 (2012), pp. 1042–1046.

[33] Dill, Ken A, Ozkan, S Banu, Shell, M Scott, and Weikl, Thomas R. "The protein folding problem". *Annu. Rev. Biophys.* 37 (2008), pp. 289–316.

[34] Eghdami, Mahdie, Dehghani, Toktam, and Naghibzadeh, Mahmoud. "BetaProbe: A probability based method for predicting beta sheet topology using integer programming". In: *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE. 2015, pp. 152–157.

[35] Eynard, B and Orantin, N. "Invariants of algebraic curves and topological expansion". *Communications in Number Theory and Physics* 1.2 (2007), pp. 347–452.

[36] Eynard, Bertrand. "A short overview of the "Topological recursion"". In: *Proceedings of ICM 2014*. IMU. 2014.

[37] Eynard, Bertrand. *Counting Surfaces*. Vol. 70. Progress in Mathematical Physics. Birkhäuser, 2016.

[38] Eynard, Bertrand, Kimura, Taro, and Ribault, Sylvain. *Random matrices*. WorkingPaper. arXiv.org, Oct. 2015.

[39] Eynard, Bertrand and Orantin, Nicolas. "Mixed correlation functions in the 2-matrix model, and the Bethe ansatz". *Journal of High Energy Physics* 2005.08 (2005), p. 028.

[40] Eynard, Bertrand and Orantin, Nicolas. "Topological expansion and boundary conditions". *Journal of High Energy Physics* 2008.06 (2008), p. 037.

[41] Eynard, Bertrand and Orantin, Nicolas. "Topological expansion of mixed correlations in the Hermitian 2-matrix model and x–y symmetry of the Fg algebraic invariants". *Journal of Physics A: Mathematical and Theoretical* 41.1 (2007), p. 015203.

[42] Eynard, Bertrand and Orantin, Nicolas. "Topological expansion of the 2-matrix model correlation functions: diagrammatic rules for a residue formula". *Journal of High Energy Physics* 2005.12 (2005), p. 034.

[43] Eynard, Bertrand and Orantin, Nicolas. "Topological recursion in enumerative geometry and random matrices". *Journal of Physics A: Mathematical and Theoretical* 42.29 (2009), p. 293001.

[44] Fonseca, Rasmus, Helles, Glennie, and Winter, Pawel. "Ranking beta sheet topologies with applications to protein structure prediction". *Journal of Mathematical Modelling and Algorithms* 10.4 (2011), pp. 357–369.

[45] Harer, J and Zagier, D. "The Euler characteristic of the moduli space of curves". *Invent. Math.* 85 (1986), pp. 457–485.

[46] Henikoff, Steven and Henikoff, Jorja G. "Amino acid substitution matrices from protein blocks". *Proceedings of the National Academy of Sciences* 89.22 (1992), pp. 10915–10919.

[47] Herschlag, Daniel, Bonilla, Steve, and Bisaria, Namita. "The story of RNA folding, as told in epochs". *Cold Spring Harbor perspectives in biology* 10.10 (2018), a032433.

[48] Huang, Fenix, Reidys, Christian, and Rezazadegan, Reza. "Fatgraph models of RNA structure". *Computational and Mathematical Biophysics* 5.1 (2017), pp. 1–20.

[49] Jumper, John, Evans, Richard, Pritzel, Alexander, Green, Tim, Figurnov, Michael, Tunyasuvunakool, Kathryn, Ronneberger, Olaf, Bates, Russ, Žídek, Augustin, Bridgland, Alex, Meyer, Clemens, Kohl, Simon A A, Potapenko, Anna, Ballard, Andrew J, Cowie, Andrew, Romera-Paredes, Bernardino, Nikolov, Stanislav, Jain, Rishub, Adler, Jonas, Back, Trevor, Petersen, Stig, Reiman, David, Steinegger, Martin, Pacholska, Michalina, Silver, David, Vinyals, Oriol, Senior, Andrew W, Kavukcuoglu, Koray, Kohli, Pushmeet, and Hassabis, Demis. "High Accuracy Protein Structure Prediction Using Deep Learning". In: *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*. 2020.

[50] Kabsch, Wolfgang. "A solution for the best rotation to relate two sets of vectors". *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32.5 (1976), pp. 922–923.

[51] Kabsch, Wolfgang and Sander, Christian. "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features". *Biopolymers* 22.12 (1983), pp. 2577–2637. DOI: 10.1002/bip.360221211.

[52] Keene, Jack D. "RNA regulons: coordination of post-transcriptional events". *Nature Reviews Genetics* 8.7 (2007), pp. 533–543.

[53] König, Julian, Zarnack, Kathi, Luscombe, Nicholas M, and Ule, Jernej. "Protein–RNA interactions: new genomic technologies and perspectives". *Nature Reviews Genetics* 13.2 (2012), pp. 77–83.

[54] Kontsevich, Maxim. "Intersection theory on the moduli space of curves and the matrix Airy function". *Communications in Mathematical Physics* 147.1 (1992), pp. 1–23.

[55] Kryshtafovych, Andriy, Barbato, Alessandro, Fidelis, Krzysztof, Monastyrskyy, Bohdan, Schwede, Torsten, and Tramontano, Anna. "Assessment of the assessment: evaluation of the model quality estimates in CASP10". *Proteins: Structure, Function, and Bioinformatics* 82 (2014), pp. 112–126.

[56] Kryshtafovych, Andriy, Monastyrskyy, Bohdan, Fidelis, Krzysztof, Schwede, Torsten, and Tramontano, Anna. "Assessment of model accuracy estimations in CASP12". *Proteins: Structure, Function, and Bioinformatics* 86 (2018), pp. 345–360.

[57] Kryshtafovych, Andriy, Prlic, Andreas, Dmytriv, Zinoviy, Daniluk, Pawel, Milostan, Maciej, Eyrich, Volker, Hubbard, Tim, and Fidelis, Krzysztof. "New tools and expanded data analysis capabilities at the Protein Structure Prediction Center". *Proteins: Structure, Function, and Bioinformatics* 69.S8 (2007), pp. 19–26.

[58] Kryshtafovych, Andriy, Schwede, Torsten, Topf, Maya, Fidelis, Krzysztof, and Moult, John. "Critical assessment of methods of protein structure prediction (CASP)—Round XIII". *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1011–1020.

[59] Loria, Andrew and Pan, T. "Domain structure of the ribozyme from eubacterial ribonuclease P." *Rna* 2.6 (1996), pp. 551–563.

[60] Moore, Melissa J. "From birth to death: the complex lives of eukaryotic mRNAs". *Science* 309.5740 (2005), pp. 1514–1518.

[61] Moult, John, Fidelis, Krzysztof, Kryshtafovych, Andriy, Schwede, Torsten, and Tramontano, Anna. "Critical assessment of methods of protein structure prediction (CASP)—round x". *Proteins: Structure, Function, and Bioinformatics* 82 (2014), pp. 1–6.

[62] Moult, John, Pedersen, Jan T, Judson, Richard, and Fidelis, Krzysztof. "A large-scale experiment to assess protein structure prediction methods". *Proteins: Structure, Function, and Bioinformatics* 23.3 (1995), pp. ii–iv.

[63] Mount, David W. "Comparison of the PAM and BLOSUM amino acid substitution matrices". *Cold Spring Harbor Protocols* 2008.6 (2008), pdb–ip59.

[64] Mulase, Motohico. "Matrix integrals and integrable systems". *Topology, geometry and field theory, K. Fukaya et al. Editors, World Scientific* (1994), pp. 111–127.

[65] Munoz, Victor and Serrano, Luis. "Intrinsic secondary structure propensities of the amino acids, using statistical ϕ–ψ matrices: comparison with experimental scales". *Proteins: Structure, Function, and Bioinformatics* 20.4 (1994), pp. 301–311.

[66] Needleman, Saul B. and Wunsch, Christian D. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of Molecular Biology* 48.3 (1970), pp. 443–453. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(70)90057-4. URL: http://www.sciencedirect.com/science/article/pii/0022283670900574.

[67] Orland, Henri and Zee, Anthony. "RNA folding and large N matrix theory". *Nuclear Physics B* 620.3 (2002), pp. 456–476.

[68] Penner, R. C. "Decorated Teichmüller Theory of Bordered Surfaces". *Commun. Anal. Geom.* 12.4 (2004), pp. 793–820.

[69] Penner, R. C. "Moduli spaces and macromolecules". *B. Am. Math. Soc.* 53.2 (2016), pp. 217–268.

[70] Penner, R. C. "The Decorated Teichmüller Space of Punctured Surfaces". *Commun. Math. Phys* 113 (1987), pp. 299–339.

[71] Penner, Robert C et al. "Perturbative series and the moduli space of Riemann surfaces". *Journal of Differential Geometry* 27.1 (1988), pp. 35–53.

[72] Penner, Robert, C., Knudsen, Micheal, Wiuf, Carsten, and Andersen, Jørgen Ellegaard. "Fatgraph models of proteins". *Communications on Pure and Applied Mathematics* 63.10 (2010), pp. 1249–1297.

[73] Penner, Robert, Andersen, Ebbe Sloth, Jensen, Jens Ledet, Kantcheva, Adriana Krassimirova, Bublitz, Maike, Nissen, Poul, Rasmussen, Anton Michael Havelund, Svane, Katrine Louise, Hammer, Bjørk, Rezazadegan, Reza, Nielsen, Niels Christian, Nielsen, Jakob Toudahl, and Andersen, Jørgen Ellegaard. "Hydrogen bond rotations as a uniform structural tool for analyzing protein architecture". *Nature Communications* 5 (2014). DOI: doi:10.1038/ncomms6803.

[74] Penner, Robert, Knudsen, Michael, Wiuf, Carsten Henrik, and Andersen, Jørgen Ellegaard. "An Algebro-Topological Description of Protein Domain Structure". *P L o S One* 6.5 (2011).

[75] Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. "Stereochemistry of polypeptide chain configurations". *J. Mol. Biol.* 7 (1963), pp. 95–99.

[76] Ray, Arjun, Lindahl, Erik, and Wallner, Björn. "Improved model quality assessment using ProQ2". *BMC bioinformatics* 13.1 (2012), p. 224.

[77] Reidys, C. M., Huang, F. W. D., Andersen, J. E., Penner, R. C., Stadler, P. F., and Nebel, M. E. "Topology and prediction of RNA pseudoknots". *Bioinformatics* 27.8 (2011), pp. 1076–1085.

[78] Richardson, Jane S. "β-Sheet topology and the relatedness of proteins". *Nature* 268.5620 (1977), pp. 495–500.

[79] Richardson, Jane S. "Handedness of crossover connections in beta sheets". *Proceedings of the National Academy of Sciences* 73.8 (1976), pp. 2619–2623.

[80] Ruczinski, Ingo, Kooperberg, Charles, Bonneau, Richard, and Baker, David. "Distributions of beta sheets in proteins with application to structure prediction". *Proteins: Structure, Function, and Bioinformatics* 48.1 (2002), pp. 85–97.

[81] Savojardo, Castrense, Fariselli, Piero, Martelli, Pier Luigi, and Casadio, Rita. "BCov: a method for predicting β-sheet topology using sparse inverse covariance estimation and integer programming". *Bioinformatics* 29.24 (2013), pp. 3151–3157.

[82] Senior, Andrew W, Evans, Richard, Jumper, John, Kirkpatrick, James, Sifre, Laurent, Green, Tim, Qin, Chongli, Žídek, Augustin, Nelson, Alexander WR, Bridgland, Alex, et al. "Improved protein structure prediction using potentials from deep learning". *Nature* (2020), pp. 1–5.

[83] Senior, Andrew W, Evans, Richard, Jumper, John, Kirkpatrick, James, Sifre, Laurent, Green, Tim, Qin, Chongli, Žídek, Augustin, Nelson, Alexander WR, Bridgland, Alex, et al. "Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13)". *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1141–1148.

[84] Shrestha, Rojan, Fajardo, Eduardo, Gil, Nelson, Fidelis, Krzysztof, Kryshtafovych, Andriy, Monastyrskyy, Bohdan, and Fiser, Andras. "Assessing the accuracy of contact predictions in CASP13". *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1058–1068.

[85] Solomatin, Sergey V, Greenfeld, Max, Chu, Steven, and Herschlag, Daniel. "Multiple native states reveal persistent ruggedness of an RNA folding landscape". *Nature* 463.7281 (2010), pp. 681–684.

[86] Staple, David W and Butcher, Samuel E. "Pseudoknots: RNA structures with diverse functions". *PLoS Biol* 3.6 (2005), e213.

[87] Sternberg, MJE and Thornton, JM. "On the conformation of proteins: The handedness of the connection between parallel β-strands". *Journal of molecular biology* 110.2 (1977), pp. 269–283.

[88] Strebel, Kurt. "Quadratic differentials". In: *Quadratic Differentials*. Springer, 1984, pp. 16–26.

[89] Subramani, Ashwin and Floudas, Christodoulos A. "β-sheet topology prediction with high precision and recall for β and mixed α/β proteins". *PloS one* 7.3 (2012).

[90] Tange, O. "GNU Parallel - The Command-Line Power Tool". *;login: The USENIX Magazine* 36.1 (Feb. 2011), pp. 42–47. DOI: `http://dx.doi.org/10.5281/zenodo.16303`. URL: `http://www.gnu.org/s/parallel`.

[91] Van Nostrand, Eric L, Pratt, Gabriel A, Shishkin, Alexander A, Gelboin-Burkhart, Chelsea, Fang, Mark Y, Sundararaman, Balaji, Blue, Steven M, Nguyen, Thai B, Surka, Christine, Elkins, Keri, et al. "Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP)". *Nature methods* 13.6 (2016), pp. 508–514.

[92] Vernizzi, Graziano, Orland, Henri, and Zee, A. "Classification and predictions of RNA pseudoknots based on topological invariants". *Physical Review E* 94.4 (2016), p. 042410.

[93] Vernizzi, Graziano, Orland, Henri, and Zee, Anthony. "Enumeration of RNA structures by matrix models". *Physical review letters* 94.16 (2005), p. 168103.

[94] Vernizzi, Graziano, Ribeca, Paolo, Orland, Henri, and Zee, A. "Topology of pseudoknotted homopolymers". *Physical Review E* 73.3 (2006), p. 031902.

[95] Wang, G. and Dunbrack, R. L. "PISCES: a protein sequence culling server". *Bioinformatics* 19 (2003), pp. 1589–1591.

[96] Won, Jonghun, Baek, Minkyung, Monastyrskyy, Bohdan, Kryshtafovych, Andriy, and Seok, Chaok. "Assessment of protein model structure accuracy estimation in CASP13: Challenges in the era of deep learning". *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1351–1360.

[97] Worth, Catherine L and Blundell, Tom L. "On the evolutionary conservation of hydrogen bonds made by buried polar amino acids: the hidden joists, braces and trusses of protein architecture". *BMC evolutionary biology* 10.1 (2010), p. 161.

[98] Zemla, Adam. "LGA: a method for finding 3D similarities in protein structures". *Nucleic acids research* 31.13 (2003), pp. 3370–3374.

[99] Zhang, Chao and Kim, Sung-Hou. "The anatomy of protein β-sheet topology". *Journal of molecular biology* 299.4 (2000), pp. 1075–1089.

Appendices

A Extension of Blosum62

In section 5, an extension of the blosum62 substitution matrix [46], which is a commonly used substitution matrix for the protein sequence alignment, was used. We give a detailed description of the extension below.

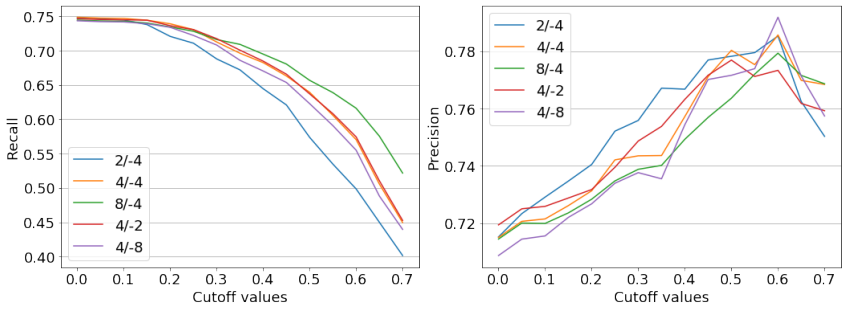
The sequences we align consists of 20 letters representing standard gene code amino acids and two extra letters, α and β. We therefore need to extend the blosum62 substitution matrix to include scores for these two extra letters. We use 4 and -4 respectively for match and mismatch involving α and β. We investigated the effect of these scores by computing average Recall and Precision for different scores as follows;

1. For each match/mismatch score combination, compute alignment scores for the first and second diagonal (i.e. for the sequences involving zero or one β-strand).

2. Let v be a number between 0 and 1. For each cell $\mathbb{P}_{(i,j)}$ in the pairing matrix $\mathbb{P}$, where the alignment scores have been computed, set the value to 1 if $\mathbb{P}_{(i,j)} > v$, 0 otherwise.

3. Compute Recall and Precision for each protein.

The average Recall and Precision for various cutoff values and score combinations are shown in figure A.1. We see the variation in Precision is very small across different score combinations. The same holds for Recall, for small cutoff values. We also note that, compared to the average Recall, the average Precision does not vary much for different cutoff values. We therefore choose the cutoff value of 0 and match/mismatch score combination of 4 and -4 for the current analysis.

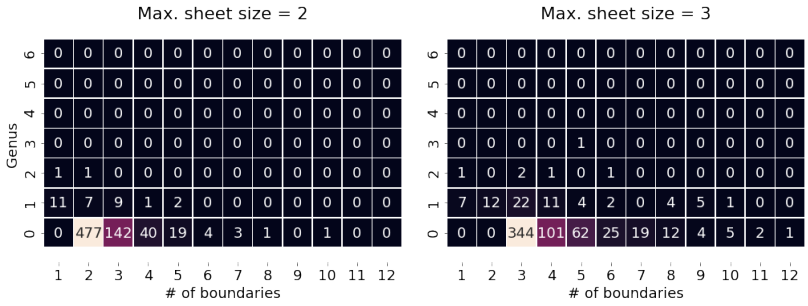


(a) Average Recall (b) Average Precision

Figure A.1: Average Recall (figure A.1a) and Precision (figure A.1b) for various cutoff values. The different lines represent different match/mismatch score combinations.

B The topology filter

The topology filter, as used in section 5, is the distribution of genera and numbers of boundary components for the protein metastructures filtered by the number of strands in the largest sheet. The filtering was done to be able to reflect the difference in the distributions between proteins with a small number of strands and those containing more strands, as one would expect those proteins with a large number of strands to have more (topologically) complex structures. The number of strands in the largest sheet was chosen as the filtering variable, because we expect the largest contribution to the genus (and the number of boundary components) to come from the largest sheet. We show the distributions up to the maximum sheet size of 10 in figure B.1. We also show the distributions of the same data, filtered by the number of sheets figure B.2 and by the number of strands figure B.3. It was thought that filtering by the number of sheets results in too few layers and will make the resulting topology filter less powerful, as it will not be able to distinguish subtler differences. To investigate whether filtering by the number of strands produces better results, we ran the binary classification (see section 5.7.1) using the topology filter with the maximum sheet size as the third axis, and one with the number of strands as the third axis. The classification results were analysed by computing the proportion of the candidate structures above certain quality thresholds, that were accepted (figure B.4). For a good filter, we expect the percentages of acceptance to increase, as we restrict to candidate structures to only look at the high-quality structures. Put another way, we expect the lines to lie diagonally from the bottom-left to top-right. It was found the filter using the maximum sheet size performed better, particularly with Recall. We did not investigate why it is the case, but it may be that folding several large sheets into energetically favourable structure is complex, and in nature a combination of one large sheet and several smaller ones is more common.



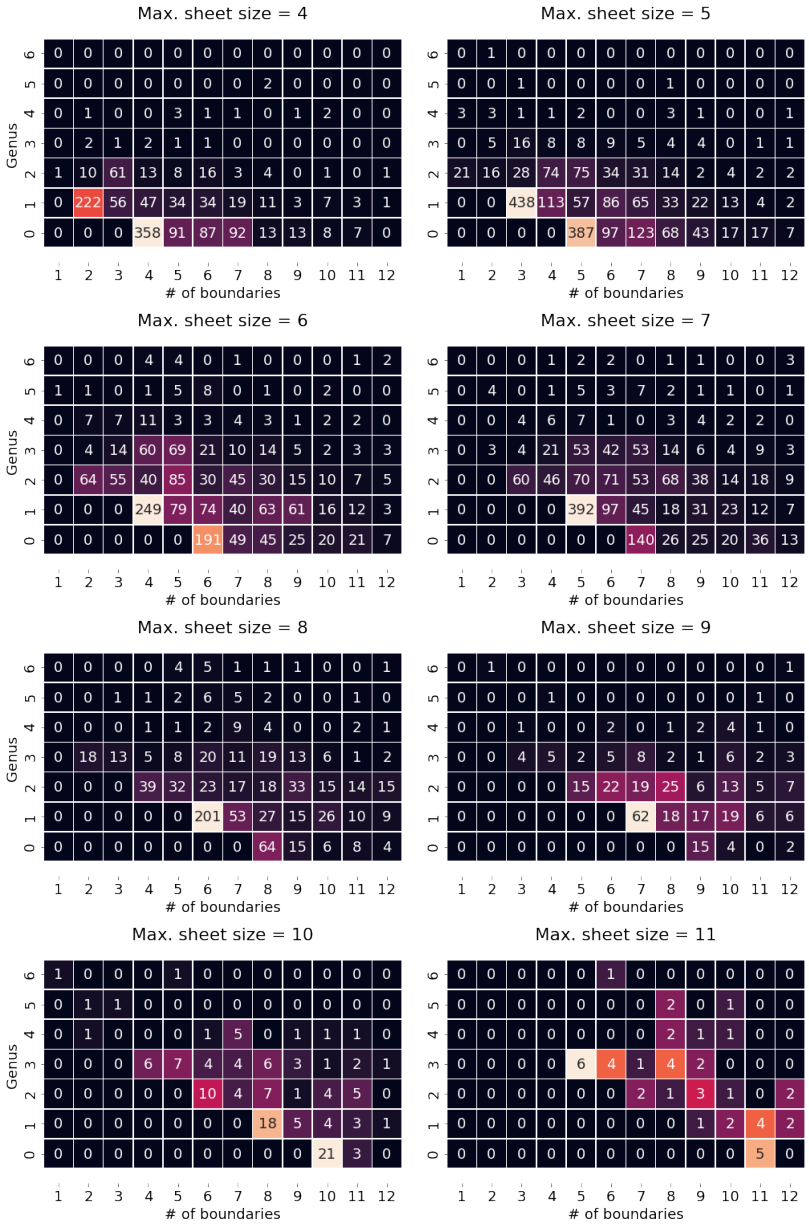


Figure B.1: The distribution of genus and number of boundary components, filtered by the size of the largest sheet.

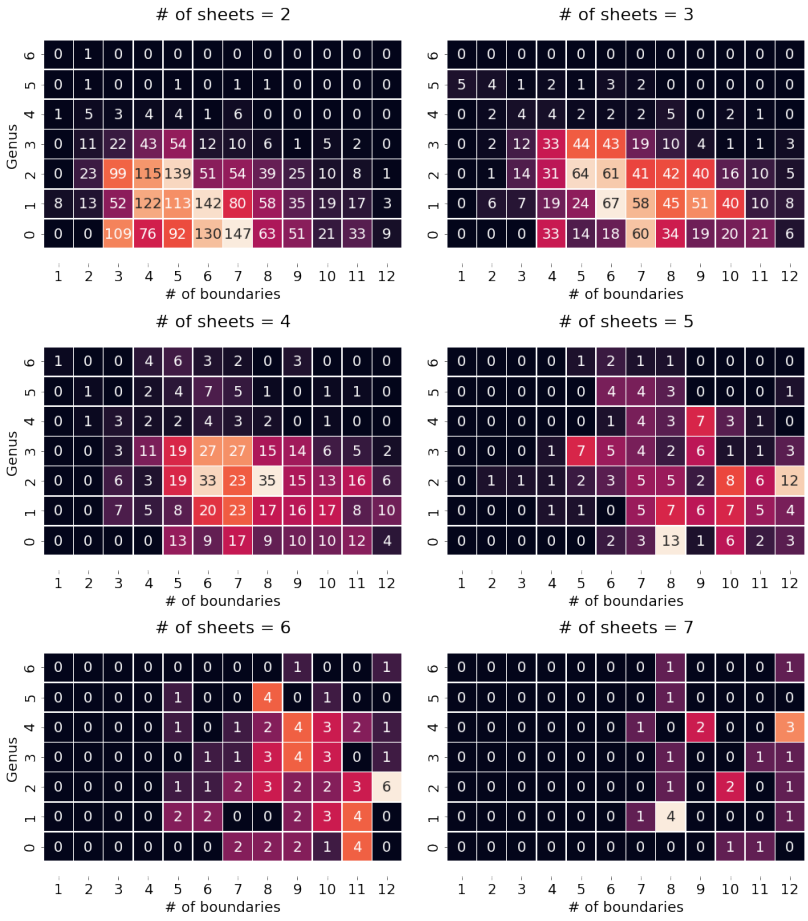
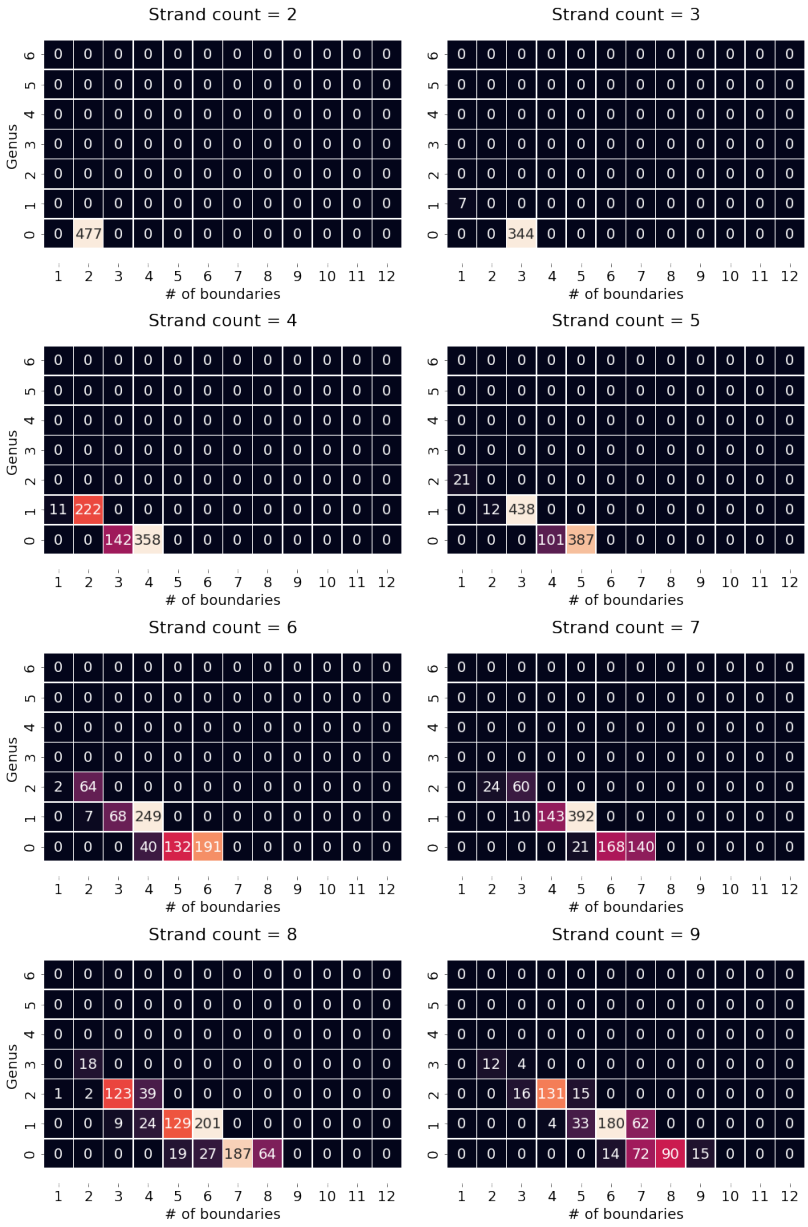


Figure B.2: The distribution of genus and number of boundary components, filtered by the number of sheets.


Strand count = 2
Strand count = 3
Strand count = 4
Strand count = 5
Strand count = 6
Strand count = 7
Strand count = 8
Strand count = 9
Genus
# of boundaries

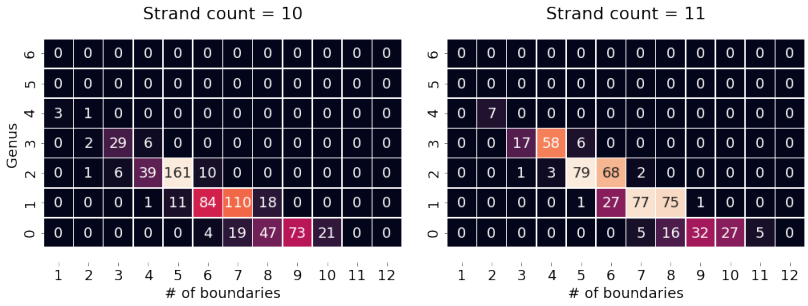
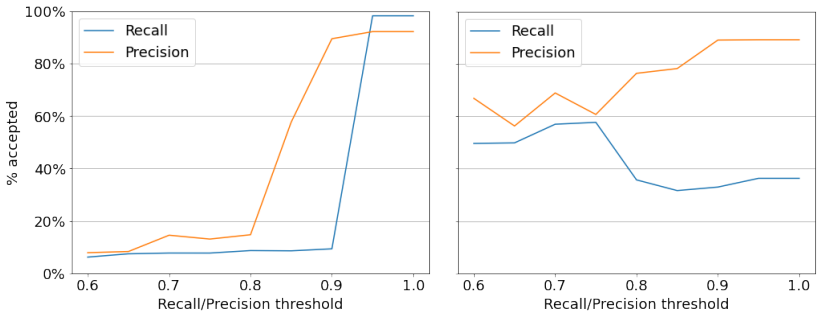


Figure B.3: The distribution of genus and number of boundary components, filtered by the size of the largest sheet.



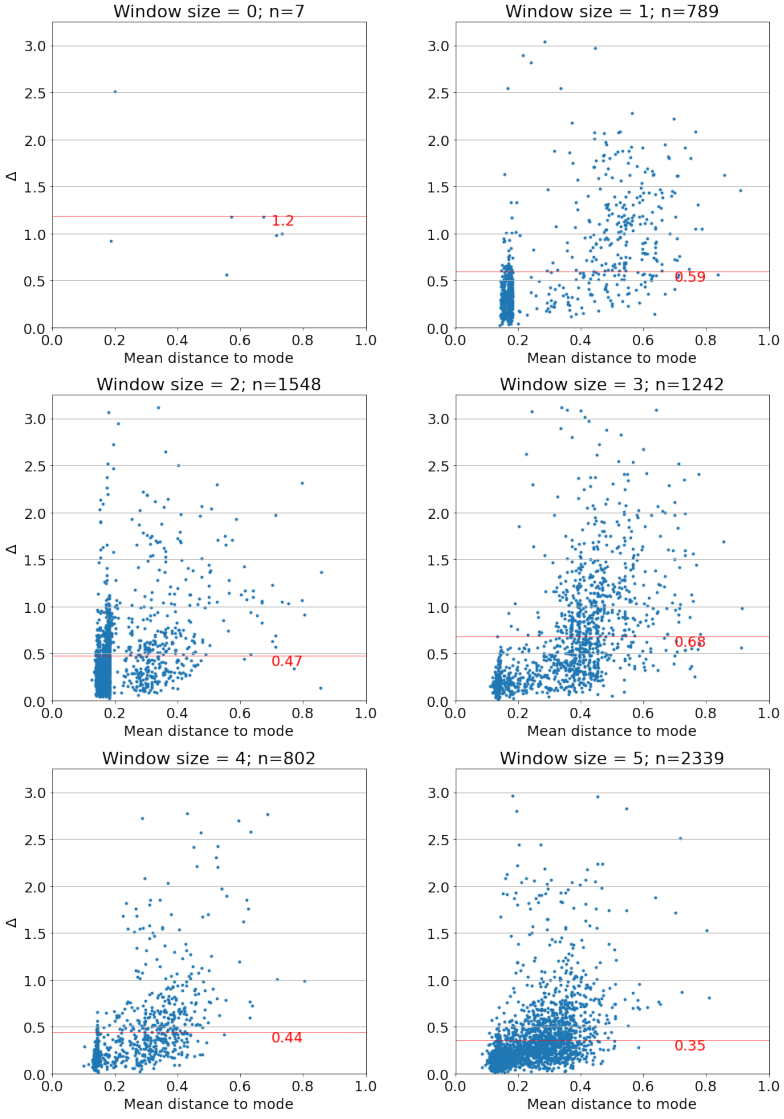
(a) Maximum sheet size

(b) Number of strands

Figure B.4: Acceptance rates for candidates above quality thresholds (measured in Recall and Precision). The topology filter with maximum sheet size (figure B.4a) shows increasing acceptance rates when the candidates are restricted to high-quality structures. On the other hand, the filter with number of strands (figure B.4b) shows relatively high acceptance rates for lower-quality candidates, and the rate drops for Recall, when the candidates are restricted to high-quality structures.

C SO(3) prediction: analysing results

The analysis of prediction results showed a number of predictions, that were made using small H-bond patterns, when a match with larger H-bond patterns may have been available (see section 7.2). As a result the score function (59) was modified to penalise the use of smaller patterns for prediction (60).



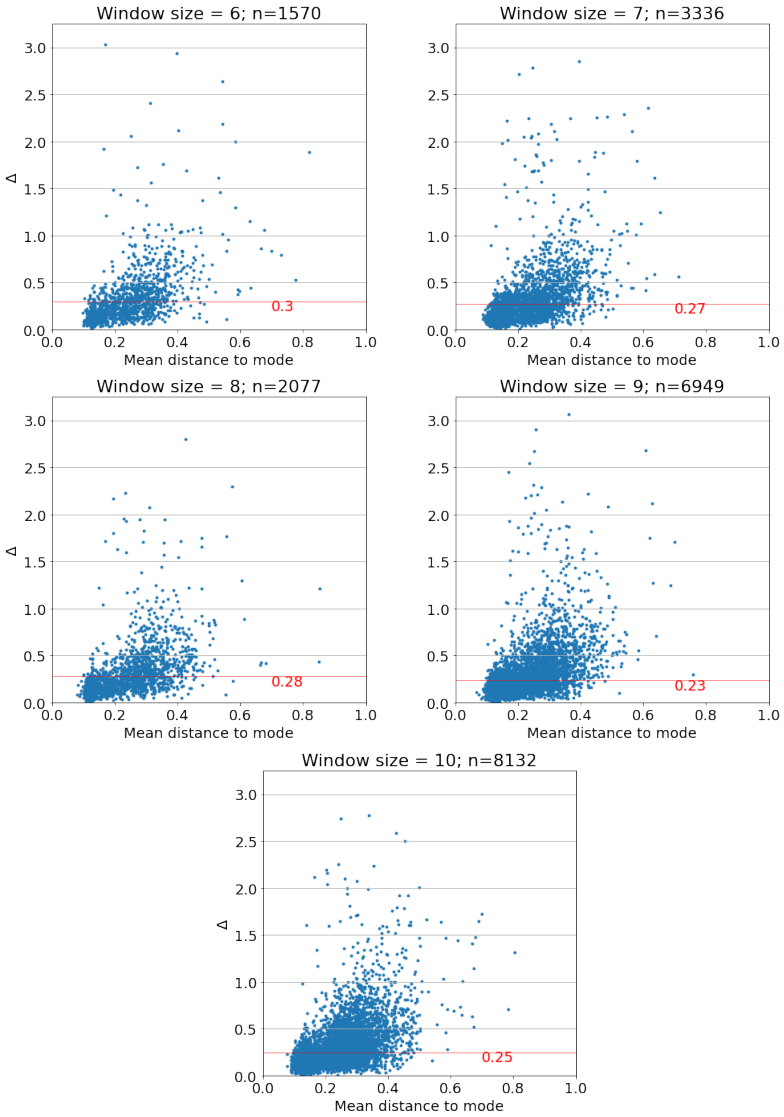


Figure C.1: Distance to the true rotation (Δ) and mean distance to cluster mode (m), filtered by the window size used. The red line indicates the mean for each plot.

D Code files

The code files used in this thesis are available from the following addresses.

1. Protein Metastructure
    - https://github.com/yukikoyanagi/python-alignment: Package for implementation of sequence alignment. Required for sequence alignment in metastructure binary classification and prediction using alignment. The code is based on work by Eser Aygün (https://github.com/eseraygun/python-alignment).
    - https://github.com/yukikoyanagi/fatgraph: Package for implementation of fatgraph. Required for metastructure analyses.
    - https://github.com/yukikoyanagi/metastructure: Scripts for protein metastructure analyses.
2. GDT algorithms
    - https://github.com/yukikoyanagi/gdt: Scripts for GDT algorithms
3. Local pattern analyses
    - https://github.com/yukikoyanagi/localpattern: Scripts for local pattern analyses

Many of the analyses were performed with GNU parallel program [90].