

Análisis configuracional de cristales moleculares empleando cuaterniones duales

Tesis que presenta

Hazel Uriel Vázquez Hernández

para obtener el grado de

Maestro en Ciencias (Química)

Asesor

Joel Ireta Moreno

Jurado:

Presidente: Dr. Herbert Höpfl

Secretario: Dr. Rodolfo Octavio Esquivel Olea

Vocal: Dr. Marcelo Enrique Galván Espinosa

Ciudad de México, 15 de mayo del 2023

Agradecimientos

Esta tesis y todo el trabajo que me ha llevado hasta aquí no hubieran sido posibles sin el apoyo de toda mi familia. En especial agradezco a mi madre, Erika, y a Juan, quienes siempre me han apoyado de manera incondicional.

A mi padre, Uriel, a quien extraño todos los días, por sus enseñanzas que me han llevado hasta este punto.

A mi esposa ilegítima, Nancy, quien ha sido mi compañera de carrera y de vida. Te agradezco por recordarme que a veces es mejor tomarse un respiro antes de continuar y por haberme ayudado a desarrollar ideas que ahora forman parte de esta tesis.

A mi asesor, Dr. Joel Ireta, por su apoyo en el desarrollo de este trabajo y por encaminarme a nuevos conocimientos.

Al Dr. Rodolfo Esquivel, por brindarme su amistad y ayudarme a desarrollarme como científico.

A mis sinodales, por su tiempo y apoyo en la revisión de este proyecto.

Al Laboratorio de Supercómputo de la UAM-I, por el tiempo de cómputo otorgado, y al CONACyT, por el apoyo económico mediante la beca de maestría (CVU: 993929).

Por último, quisiera agradecer a mis mascotas, Idril, Sally y Aragorn. Bellos seres que sin una palabra me han enseñado que la comida, el dormir y tomarse descansos también son importantes. Ellos hicieron todo el trabajo más ameno.

Índice general

1	Introducción	23
2	Marco teórico	27
2.1	Estudios computacionales del polimorfismo	27
2.2	Métodos de estructura electrónica	30
2.2.1	Métodos multielectrónicos y Hartree-Fock	30
2.2.2	Funcionales de la densidad	33
2.2.3	Sistemas periódicos	38
2.3	Métodos de agrupamiento	43
2.3.1	<i>K-means</i>	44
2.3.2	Métodos espectrales	46
2.3.3	Propagación de afinidades	48
2.3.4	Ordenamiento de puntos para identificar la estructu- ra de clústeres	50
2.3.5	Validación de las agrupaciones	52
2.4	Métricas en el espacio configuracional	54
2.4.1	Cuaterniones duales y su aplicación en el desplaza- miento de cuerpos rígidos	55
2.4.2	Raíz del error cuadrático medio (RMSD)	64
2.4.3	Descriptor de coordenadas relativo	65
2.5	Cristales moleculares analizados	67
2.6	Generación aleatoria de estructuras cristalinas moleculares . .	70
3	Hipótesis	73
4	Objetivos general y particulares	73
5	Metodología	74

5.1	Generación del conjunto inicial de estructuras cristalinas	77
5.2	Análisis energético de las estructuras cristalinas	79
5.3	Análisis configuracional con cuaterniones duales	80
5.3.1	Análisis de coordenadas de sistemas periódicos	80
5.3.2	Caso particular: estructuras con diferente número de moléculas en la celda unitaria	82
5.3.3	Generación de cuaterniones duales	84
5.4	Comparación de las configuraciones y energías de estructuras cristalinas moleculares	87
5.5	Agrupamiento	96
5.6	Análisis de las métricas y de las agrupaciones	99
6	Resultados	102
6.1	Verificación de los algoritmos	103
6.2	Análisis de métodos de agrupamiento	108
6.3	Análisis de las métricas para medir la disimilitud entre confi- guraciones cristalinas	112
6.4	Aplicación del método de agrupamiento espectral	122
6.5	Análisis de sistemas polimorficos	136
7	Conclusiones	142
8	Perspectivas	145
9	Anexo A	160
9.1	Análisis adicionales	160
10	Anexo B	166
10.1	Demostración del cambio de referencia	166
10.2	Demostración de la simetría de las medidas s y z	168
11	Anexo C	171
11.1	Código para la generación de matrices de distancia	171
11.1.1	Cálculo de las medidas z y s	171
11.1.2	Cálculo de la medida $RMSD$	194

11.1.3	Cálculo de la medida RCD	196
--------	--------------------------------------	-----

Glosario e índice de abreviaturas

bmm	Búsqueda del máximo mínimo, algoritmo de búsqueda de mínimos de una matriz por renglón y por columna
bmp	Búsqueda del mínimo promedio, algoritmo de búsqueda de mínimos globales de una matriz
Car	Cardinalidad, número de elementos en un conjunto
configuración	Posiciones y orientaciones de un conjunto de moléculas en relación a una molécula de referencia que definen una estructura cristalina
clúster o agrupación	Conjunto de objetos relacionados entre sí. En particular, este término se emplea en esta tesis para referirse a un conjunto de estructuras cristalinas con configuraciones o con energías electrónicas similares
CSD	Del inglés <i>Cambridge Structural Database</i>
CV(RMSD)	Coefficiente de variación o normalización de la raíz de la desviación cuadrática media.
$D_{l,c}^a$	Matriz de distancia que contiene la comparación de todos los pares posibles de redes cristalinas analizadas, construida con la medida a , el algoritmo de búsqueda de mínimos c y considerando un límite en la búsqueda de mínimos l
DBSCAN	Del inglés D ensity- B ased S patial C lustering of A pplications with N oise
DFT	Del inglés <i>Density Functional Theory</i>
hiperparámetro	Dato de entrada que se emplea en métodos de aprendizaje automático, el cual se mantiene constante durante la ejecución de los algoritmos y su estimación debe ser realizada por el usuario
LDA	Del inglés <i>Local Density Approximation</i>
<i>medoide</i>	Estructura cristalina más cercana al centro de un clúster
<i>métrica</i>	Función que define una distancia entre cada par de elementos de un conjunto. Esto incluye el concepto intuitivo de distancia entre dos puntos en el espacio tridimensional, pero esta definición se extiende a comparaciones que pueden depender de una mayor cantidad de variables

MSD	Del inglés M ean S quare D eviation
N	Número total de moléculas obtenidas con un número finito de traslaciones periódica de la celda unitaria que representan una estructura cristalina
N_s	Número de estructuras cristalinas generadas de manera aleatoria para un compuesto molecular
OPTICS	Del inglés O rdering P oints to I dentify the C lustering S tructure
PAW	Del inglés <i>Projected Augmented Waves</i>
PBE	P erdew- B urke- E rnzerhof
RCD	Del inglés <i>Relative Coordinate Descriptor</i>
RMSD	Raíz cuadrada de la desviación cuadrática media, del inglés <i>Root Mean Square Deviation</i>
t.q.	Tal que
Z	Número de moléculas en la celda unitaria de una estructura cristalina
ϵ	Hiperparámetro del método de agrupamiento <i>OPTICS</i> que delimita los datos asociados a cada clúster

Índice de figuras

1	Representación esquemática de la rotación y traslación de un cuerpo rígido, donde $\hat{t}$ es el vector de rotación, θ el ángulo de rotación y $\vec{T}$ el vector entre los centros de coordenadas. El cuerpo rígido es representado por las triadas de vectores en naranja y azul.	60
2	Representación esquemática de la descripción vectorial de moléculas en redes cristalinas propuesta con el descriptor de coordenadas relativo, donde (a) representa el vector de posición, $\vec{T}_i$, y (b) el vector de orientación relativa, $\vec{Q}_i$	66
3	Representaciones 3D de las geometrías moleculares de (a) formamida, (b) ácido oxálico e (c) hidrazida maleica.	67
4	Esquema general de la metodología desarrollada en esta tesis. Las líneas rectas indican información sobre el trabajo adicional realizado para cumplir con lo planteado en el esquema principal señalado con flechas. Los detalles sobre cada punto planteado en el esquema se presentan en las siguientes secciones.	74
5	Diagrama de flujo para definir el límite de traslaciones periódicas de la celda central, z_p , tal que el número de moléculas que representen una estructura cristalina, M , generadas al trasladar periódicamente la celda unitaria, sea mayor o igual al número de moléculas a comparar entre cada par de estructuras, l	83

6	Diagrama de flujo del cálculo de cuaterniones duales para un conjunto de estructuras cristalinas de un compuesto químico molecular. Sea <i>alg</i> un parámetro que determina el tipo de algoritmo de búsqueda de mínimos a emplear (<i>bmp</i> o <i>bmm</i> , descritos en la siguiente sección). . .	86
7	Diagramas de flujo mostrando los algoritmos de búsqueda de valores mínimos (a) <i>bmp</i> y (b) <i>bmm</i> . En el caso del algoritmo <i>bmm</i> se muestra solo el caso de la búsqueda por renglones; sin embargo, este algoritmo es equivalente a la búsqueda por columnas al invertir los índices de la matriz D	92
8	Diagrama de flujo mostrando la metodología empleada para la comparación de redes cristalinas moleculares generadas de forma aleatoria para los diferentes compuestos estudiados en esta tesis. Sea <i>alg</i> un parámetro que determina el tipo de algoritmo de búsqueda de mínimos a emplear: <i>bmp</i> , para el cual se consideran dos variaciones (1 y 3), y <i>bmm</i> (2).	94
9	Esquema mostrando los nodos o moléculas cercanas a la molécula de referencia en la celda central al replicar celdas unitarias con (a) Z=2 y (b) Z=4 moléculas, considerando que $ \vec{c} < \vec{a} , \vec{b} $. Se resalta en rojo la molécula de referencia de la celda central y en líneas verdes discontinuas se muestran sus vecinos cercanos en dos dimensiones. . .	95
10	Diagrama de flujo del algoritmo desarrollado para el agrupamiento de un conjunto de estructuras cristalinas de un mismo compuesto molecular con base en la comparación por pares de sus configuraciones o de sus energías a través de matrices de distancia, D , basado en los métodos de agrupamiento espectrales.	98

- 11 Diferencia en porcentaje de las etiquetas asignadas a las estructuras cristalinas de la formamida agrupadas con el método de agrupamiento espectral a partir de las matrices D_{bmm}^z calculadas con el algoritmo *bmm* compilado con *gfortran* y con *ifortran* para conjuntos de estructuras cristalinas de la formamida de diferente tamaño. Las matrices se calcularon por cada compilador con el número de moléculas cercanas a la molécula de referencia igual a 14 (azul), 26 (naranja), 32 (verde) y 107 (rojo). 106

- 12 Coeficiente de silueta promedio de los clústeres obtenidos con el método de propagación de afinidades utilizando diferentes valores del hiperparámetro de *preferencia* empleando las matrices (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ de un subconjunto de 3000 estructuras cristalinas de la formamida generadas aleatoriamente. 108

- 13 Coeficientes de silueta promedio obtenidos en la variación del número de clústeres al emplear las matrices de distancia (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ y, como método de agrupamiento, el método espectral seguido de *k-means* de un subconjunto de 3000 estructuras cristalinas de la formamida. 109

- 14 Coeficiente de silueta promedio máximo (respecto a los valores obtenidos para diferentes números de clústeres) para diferentes valores de σ al emplear las matrices de distancia (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ y el método espectral seguido de *k-means* para un subconjunto de 3000 estructuras cristalinas de la formamida. 110

- 15 *RMSD* de 106 moléculas de una de las estructuras cristalinas de la formamida, generadas aleatoriamente, respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en relación con las métricas (a) *z*, (b) *s* y (c) *RCD*. En el caso de la gráfica (a) se muestra el valor cuadrado del *RMSD*, sea el *MSD*, con el fin de mostrar de manera más clara la relación entre esta medida y *z*. 112

- 16 Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^z$ que fueron obtenidos con el algoritmo *bmp* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. En la figura (d) se muestran con elipses discontinuas en color rojo tres regiones separadas de puntos de manera aproximada que podrían representar clústeres de las estructuras cristalinas de la formamida. 114

- 17 Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^z$ obtenidos con el algoritmo *bmm* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para ambas matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. En la figura (d) se muestran con elipses discontinuas en color rojo tres regiones separadas de puntos de manera aproximada que podrían representar clústeres de las estructuras cristalinas de la formamida. 115

- 18 Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{RMSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^{RCD}$ obtenidos con el algoritmo *bmp* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para ambas matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. 117

- 19 Métricas $\mathbf{D}^z$, $\mathbf{D}^s$ y $\mathbf{D}^{RMSD}$ entre los polimorfos (a) $MH1 - MH2$, (b) $MH1 - MH3$, (c) $MH2 - MH3$ y (d) $OA\alpha - OA\beta$ calculadas con el algoritmo *bmm* considerando diferentes valores de moléculas vecinas (l) en relación a la molécula de referencia. Para este análisis se emplearon los datos de las estructuras cristalinas experimentales reportados en la CSD (referencias en la tabla 1). 118

20	Métricas $\mathbf{D}^z$, $\mathbf{D}^s$ y $\mathbf{D}^{RMSD}$ entre los polimorfos (a) $MH1 - MH2$, (b) $MH1 - MH3$, (c) $MH2 - MH3$ y (d) $OA\alpha - OA\beta$ calculadas con el algoritmo <i>bmp</i> considerando diferentes valores de moléculas vecinas (l) en relación a la molécula de referencia. Para este análisis se emplearon los datos de las estructuras cristalinas experimentales reportados en la CSD (referencias en la tabla 1). Los valores de la matriz $\mathbf{D}_{bmp}^{RCD}$ se presentan de manera separada debido a que su escala es mucho mayor a la del resto de las métricas.	119
21	Coeficientes de silueta promedio de las agrupaciones obtenidas con las métricas z , $RMSD$ y RCD para números diferentes de moléculas vecinas (l) en relación a la molécula de referencia, considerados al calcular las matrices de distancia con los algoritmos (a) <i>bmp</i> y (b) <i>bmm</i> para un subconjunto de 3000 estructuras cristalinas de la formamida; en todos los casos se usó el método de agrupamiento espectral.	123
22	Números de clústeres óptimos (generan los valores más altos del puntaje de silueta promedio) obtenidos con las métricas z , $RMSD$ y RCD para diferentes valores de moléculas cercanas (l) en relación a la molécula de referencia, considerados al calcular las matrices de distancia con los algoritmos (a) <i>bmp</i> y (b) <i>bmm</i> para un subconjunto de 3000 estructuras de la formamida; en todos los casos se usó el método de agrupamiento espectral.	125
23	Coeficientes de silueta promedio óptimos de los clústeres generados con las métricas z y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) <i>bmp</i> y (b) <i>bmm</i> para subconjuntos de estructuras cristalinas de la formamida de diferente tamaño; en todos los casos se usó el método de agrupamiento espectral. Además, en morado se muestran los valores correspondientes a generar las agrupaciones de manera aleatoria considerando solo 2 clústeres y calcular el coeficiente de silueta con la matriz $\mathbf{D}_{107,bmp}^z$; en café se muestran los valores para las agrupaciones obtenidas considerando la diferencia de energía electrónica entre todos los pares de estructuras cristalinas del conjunto analizado ($\Delta E_{\alpha,\beta} = E^\alpha - E^\beta \text{ kcal/mol}$).	126

- 24 Histogramas de las energías de las estructuras cristalinas generadas aleatoriamente para la formamida. El histograma en azul considera la energía electrónica de todas las estructuras cristalinas (15594); en naranja y verde se muestran las energías de los dos clústeres de estructuras identificados con la matriz $\mathbf{D}^{\Delta E}$. Además, para cada caso se muestra el valor promedio, μ , y la desviación estándar, σ , en *kcal/mol*. 129
- 25 Coeficiente de silueta para cada estructura cristalina de la formamida en los clústeres obtenidos con las matrices de distancia (a) $D_{14,bmp}^z$, (b) $D_{26,bmp}^z$, (c) $D_{107,bmp}^z$, (d) $D_{14,bmm}^z$, (e) $D_{26,bmm}^z$, (f) $D_{107,bmm}^z$, (g) D_8^{RCD} y (h) $D^{\Delta E}$. Así, las primeras dos filas corresponden a la medida z , la primera con el algoritmo *bmp* y la segunda con *bmm*, los valores de l aumentan de izquierda a derecha. El coeficiente de silueta promedio se muestra con la línea roja discontinua. 133
- 26 Coeficiente de silueta promedio de los clústeres obtenidos con las métricas z , *RMSD* y *RCD* para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (MH); en todos los casos se usó el método de agrupamiento espectral. 136
- 27 Coeficiente de silueta promedio de los clústeres obtenidos con las métricas z , *RMSD* y *RCD* para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 970 estructuras cristalinas del ácido oxálico (OA); en todos los casos se usó el método de agrupamiento espectral. 137
- 28 Coeficientes de silueta para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (MH) de acuerdo con los clústeres obtenidos con las matrices de distancia (a) $D_{107,bmp}^z$, (b) $D_{107,bmm}^z$ y (c) D_8^{RCD} . El coeficiente de silueta promedio se muestra con la línea roja discontinua. 138

29	Distancia promedio entre los vectores que definen la celda unitaria de las estructuras determinadas experimentalmente (a) <i>MH1</i> , (b) <i>MH2</i> , (c) <i>MH3</i> , (d) <i>OA</i> – α y (e) <i>OA</i> – β y de las estructuras cristalinas generadas aleatoriamente que fueron identificadas como similares con las métricas <i>z</i> , <i>RMSD</i> y <i>RCD</i> . En el caso de las medidas <i>z</i> y <i>RMSD</i> se emplearon los algoritmos <i>bmp</i> y <i>bmm</i> . Todas las matrices de distancia se calcularon considerando $l=8, 14, 26$ y 107	140
30	<i>RMSD</i> de 106 moléculas de una de las estructuras de la hidrazida maleica respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en relación con las métricas (a) <i>z</i> , (b) <i>s</i> y (c) <i>RCD</i> de las mismas moléculas. En el caso de la gráfica (a) se muestra el valor cuadrado del <i>RMSD</i> , sea el <i>MSD</i> , con el fin de mostrar de manera más clara la relación entre esta medida y <i>z</i>	160
31	<i>RMSD</i> de 106 moléculas de una de las estructuras del ácido oxálico respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en el mismo sistema en relación con las métricas (a) <i>z</i> , (b) <i>s</i> y (c) <i>RCD</i> de las mismas moléculas. En el caso de la gráfica (a) se muestra el valor cuadrado del <i>RMSD</i> , sea el <i>MSD</i> , con el fin de mostrar de manera más clara la relación entre esta medida y <i>z</i>	160
32	Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^s$ obtenidos con el algoritmo <i>bmp</i> para (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (<i>l</i>) en relación a la molécula de referencia (<i>l</i>). Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos.	161

- 33 Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^s$ obtenidos con el algoritmo *bmm* para (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia (l). Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. 162
- 34 Coeficiente de silueta promedio de los clústeres obtenidos con la medida z y su raíz cuadrada (z_{rc}) considerando 107 moléculas vecinas a la molécula de referencia (l) al calcular las matrices de distancia con el algoritmo *bmp* para subconjuntos de diferentes tamaños de estructuras cristalinas de la formamida; en todos los casos se usó el método de agrupamiento espectral. 163
- 35 Número de clústeres óptimos obtenidos con las medidas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (*MH*); en todos los casos se usó el método de agrupamiento espectral. 164
- 36 Número de clústeres óptimos obtenidos con las medidas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 970 estructuras cristalinas del ácido oxálico (*OA*); en todos los casos se usó el método de agrupamiento espectral. 164
- 37 Coeficientes de silueta para cada estructura en los clústeres obtenidos con las matrices de distancia (a) $D_{107,bmp}^z$, (b) $D_{107,bmm}^z$ y (c) D_8^{RCD} , de un conjunto de 970 estructuras cristalinas del ácido oxálico (*OA*). El coeficiente de silueta promedio se muestra con la línea roja discontinua. 165

38	Diferencia relativa en porcentaje entre el volumen de la celda unitaria de las estructuras determinadas experimentalmente (a) <i>MH1</i> , (b) <i>MH2</i> , (c) <i>MH3</i> , (d) <i>OA</i> – α y (e) <i>OA</i> – β y las estructuras cristalinas identificadas como similares con las métricas <i>z</i> , <i>RMSD</i> y <i>RCD</i> . En el caso de las medidas <i>z</i> y <i>RMSD</i> se emplearon los algoritmos <i>bmp</i> y <i>bmm</i> . Todas las matrices de distancia se calcularon considerando $l = 8, 14, 26$ y 107	165
----	---	-----

Índice de tablas

- 1 Datos cristalográficos de los compuestos analizados y considerados como sistemas de referencia para la implementación y validación de la metodología desarrollada en esta tesis. Los polimorfos del ácido oxálico están etiquetados como *OA* y los de la hidrazida maleica como *MH*. Se indican las claves de referencia en la base de datos CSD de los archivos empleados en este trabajo. 69
- 2 Valores del número máximo de estructuras por grupo espacial (compatible con Z), N_{max} , del volumen promedio, $\bar{V}$, de la desviación estándar del volumen ($\sigma_V = \bar{V}/2$) y del número de moléculas, Z , de las celdas unitarias que fueron empleados para generar N_s estructuras cristalinas aleatoriamente con el programa *Genarris* por cada estructura cristalina conocida experimentalmente para los compuestos analizados en esta tesis. El volumen promedio y el número de moléculas en la celda unitaria se tomaron de los datos experimentales reportados en la tabla 1. Las estructuras cristalinas generadas por cada polimorfo del ácido oxálico y de la hidrazida maleica fueron conjuntadas de manera aleatoria, tal que el número total de estructuras analizadas fue de 970 y 1092, respectivamente. 78

3	Valores promedio y máximo del $RMSD$ de las distancias entre las coordenadas de todos los átomos de las moléculas obtenidas al generar el ambiente químico alrededor de la celda unitaria y las que resultan de aplicar los cuaterniones duales correspondientes sobre la molécula de referencia en cada estructura cristalina generada de manera aleatoria para cada compuesto analizado en esta tesis. El valor promedio del $RMSD$ se calculó primero sobre las M moléculas en la primera zona periódica de cada estructura y los valores se promediaron a su vez sobre todas las redes cristalinas (N_s). El valor máximo se obtuvo considerando todas las transformaciones y todas las estructuras cristalinas en el conjunto generado aleatoriamente.	104
4	Análisis energético de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida, que se generaron aleatoriamente, con la matriz de distancia $\mathbf{D}^{\Delta E}$. Para cada clúster se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas; y la población relativa (PR). . . .	129
5	Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con las matrices de distancia D_{bmp}^z que fueron calculadas con 14, 26 y 107 moléculas cercanas (l) en relación a la molécula de referencia. Para los clústeres obtenidos se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los medoides ($CV(RMSD)$).	130
6	Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con las matrices de distancia D_{bmm}^z que fueron calculadas con 14, 26 y 107 moléculas cercanas (l) en relación a la molécula de referencia. Para los clústeres obtenidos se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los medoides ($CV(RMSD)$).	131

7	Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con la matriz de distancia $\mathbf{D}_{8,bmp}^{RCD}$. Para cada clúster se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los <i>medoides</i> ($CV(RMSD)$).	132
---	--	-----

Resumen

Los cambios en el acomodo espacial (configuración) de las moléculas en las redes cristalinas de polimorfos llevan a materiales o fármacos con propiedades físicas y químicas distintas entre sí. Por lo tanto, resulta de gran importancia el desarrollo de metodologías teórico-computacionales que permitan predecir las diferentes estructuras cristalinas que un compuesto puede adoptar. Para esto, suele proponerse un gran conjunto de estructuras cristalinas del compuesto de interés. Dado que no es computacionalmente factible optimizar energéticamente todas las estructuras que se propongan, es necesario contar con herramientas que permitan delimitar las estructuras cristalinas relevantes, lo cual puede hacerse en términos de las configuraciones de las moléculas en las redes cristalinas.

En esta tesis se estableció el objetivo de determinar clústeres de estructuras cristalinas con configuraciones moleculares similares utilizando como criterios de distinción medidas de disimilitud entre las orientaciones de moléculas en relación a una molécula de referencia en redes cristalinas. Para esto se generaron estructuras cristalinas aleatoriamente para formamida, hidrazida maleica y ácido oxálico. Las estructuras cristalinas fueron descritas y comparadas por pares empleando conjuntos de cuaterniones duales, herramientas matemáticas que permitieron condensar la información de cómo se distribuyen y orientan las moléculas en las redes cristalinas. Con base en lo anterior y empleando métodos de agrupamiento se generaron clústeres de estructuras cristalinas con configuraciones moleculares similares. La descripción de las estructuras cristalinas generada con los cuaterniones duales fue comparable a la generada con otras métricas reportadas en la literatura. Sin embargo, esta no fue lo suficientemente robusta para generar una subdivisión en clústeres de las estructuras cristalinas de los compuestos analizados que pudiese ser útil para la predicción de la estructura cristalina de estos compuestos.

Abstract

The changes in the spatial arrangement (configuration) of molecules in the crystal lattices of polymorphs lead to materials or drugs with different physical and chemical properties from each other. Therefore, the development of theoretical-computational methodologies that allow predicting the different crystal structures that a chemical compound can adopt is of great importance. For this, a large set of crystal structures of the compound of interest is usually proposed. Since it is not computationally feasible to energetically optimize all the proposed structures, it is necessary to have tools that allow delimiting the relevant crystal structures, which can be done in terms of the configurations of the molecules in the crystal lattices.

In this thesis, the objective was established to determine clusters of crystalline structures with similar molecular configurations using dissimilarity measures between the orientations of molecules in relation to a reference molecule in crystalline lattices as distinguishing criteria. For this purpose, crystalline structures were randomly generated for formamide, maleic hydrazide and oxalic acid. The crystalline structures were described and compared pairwise using sets of dual quaternions, mathematical tools that allowed the information on how molecules are distributed and oriented in crystalline lattices to be condensed. Based on the above and using clustering methods, clusters of crystalline structures with similar molecular configurations were generated. The description of the crystalline structures generated with dual quaternions was comparable to that generated with other metrics reported in the literature. However, it was not robust enough to generate a subdivision into clusters of the crystalline structures of the analyzed compounds that could be useful for predicting the crystalline structure of these compounds.

1. Introducción

El estudio de la relación estructura-propiedades de sustancias químicas ha sido fundamental en el desarrollo de la ciencia y tecnología. La estructura interna de una sustancia, una molécula o un sólido, determina su comportamiento físico y químico. Sin embargo, esta relación es multifactorial y puede ser difícil describirla. Un claro ejemplo de esto es el polimorfismo, evidenciado por Wöhler y Liebig (1832). Este término se refiere a la capacidad de una sustancia de adquirir más de una estructura cristalina, cada una de ellas con propiedades distintas. Este fenómeno es de gran relevancia en química, ya que es muy frecuente en compuestos orgánicos (Stahly 2007). La formación de diferentes tipos de estructuras cristalinas para un mismo compuesto puede depender de diferentes factores al momento de la cristalización como el solvente, la temperatura, impurezas y, en el caso de sólidos moleculares, la geometría molecular. Así, en el desarrollo de fármacos, por ejemplo, es realmente importante conocer las posibles estructuras cristalinas que puede adoptar el medicamento sintetizado con el fin de seleccionar el método de síntesis adecuado, ya que de esto depende la eficacia y estabilidad del fármaco (Bernstein 2002). Asimismo, el polimorfismo es de gran importancia en el desarrollo de nuevos materiales, puesto que las diferentes estructuras cristalinas de un mismo compuesto pueden llevar a propiedades muy distintas (Pfattner et al. 2010, Yan y Evans 2014). Por esta razón, se han desarrollado múltiples técnicas computacionales que permiten predecir la estructura cristalina de sólidos moleculares conociendo únicamente la estructura molecular; sin embargo, las predicciones no siempre son correctas y en algunos casos pueden llegar a ser computacionalmente inviables.

La predicción de estructuras cristalinas de moléculas consiste esencialmente en filtrar un conjunto grande de estructuras posibles en función de sus energías. Para esto se deben cumplir principalmente dos condiciones: se debe tener una forma correcta de determinar la energía de cada una de las estructuras posibles y se debe tener un procedimiento de búsqueda adecuado que permita identificar las estructuras más estables en las hipersuperficies de energía generadas (Price 2004, 2008). Para cumplir la primera condición pueden utilizarse cálculos de estructura electrónica en sólidos, lo cual es actualmente computacionalmente viable para sistemas relativamente grandes, o pueden utilizarse campos de fuerza parametrizados de manera empírica para interacciones “intra-” e “inter-” moleculares (Li et al. 2018, McDonagh et al. 2019). Sin embargo, en ambos casos, muchas veces se desprecian los efectos entrópicos y térmicos. Es decir, la mayor parte de los desarrollos en este tema han sido para mejorar los campos de fuerzas utilizados y asumen que la estructura estable a 0 K lo

sigue siendo a temperaturas más altas. Sin embargo, la contribución de la entropía y las correcciones térmicas a la energía pueden ser un factor determinante conforme aumenta la temperatura. Aun así, la manera de determinar, en particular, la corrección entrópica para estos sistemas tampoco es evidente. Actualmente, se han tenido grandes avances en la predicción de estructuras polimórficas (Reilly 2016), pero aún no existe un método universal.

La entropía de los sólidos, como se mencionó previamente, es fundamental si se busca mejorar la predicción de estructuras cristalinas estables. Además, la diferencia en energía reticular y vibracional, obtenida de manera computacional, entre polimorfos es pequeña, usualmente por debajo de 2 kJ/mol (Nyman y Day 2015, Cruz-Cabeza et al. 2015). Así, un análisis únicamente energético de posibles estructuras cristalinas puede no ser suficiente para identificar polimorfos. Sin embargo, la diferencia en entropía para este tipo de estructuras puede ser considerable conforme la temperatura aumenta (Nyman y Day 2015). Por lo tanto, la entropía puede ser un factor determinante al momento de analizar sistemas que presentan polimorfismo; aun así, la manera de determinar esta propiedad para los cristales no es única (Piaggi y Parrinello 2018, Nyman y Day 2015). La entropía podría, en principio, obtenerse utilizando dinámica molecular estudiando la transición líquido-sólido. Sin embargo, el tiempo en el que ocurre la cristalización es mucho mayor a las escalas de tiempo posibles computacionalmente. Por lo tanto, se han tenido que buscar maneras de aproximar la entropía. Una de las aproximaciones más usuales ha sido considerar que la entropía en sólidos es únicamente configuracional; es decir, se considera sólo la contribución entrópica asociada a las orientaciones y posiciones de los átomos o moléculas en las redes cristalinas. Así, se han propuesto modelos para calcular esta propiedad como los denominados cuasi-químicos que consideran la formación de los cristales por pares de átomos o moléculas (Bhadeshia s.f.); estos mismos modelos han sido extendidos considerando grupos o clústeres, lo cual es conocido como métodos de variación de cúmulos (CVM por sus siglas en inglés) (Kikuchi 1951, Sanchez et al. 1984). Otros modelos propuestos se basan en utilizar metadinámicas, las cuales consisten en dinámicas moleculares que utilizan funciones de peso para favorecer ciertos eventos de interés como la cristalización. En particular, se ha propuesto que en estas funciones de peso se incluya la entropía orientacional (basada en funciones de distribución radial) (Piaggi y Parrinello 2018).

Por otra parte, recientemente se ha planteado utilizar la entropía de Shannon (Shannon 1948) de manera equivalente a la entropía termodinámica. Esta medida usada inicialmente para optimizar la codificación de mensajes se ha extendido a otras áreas. En particular, se ha encontrado su equivalencia con la entropía de Bol-

tzmann (Schwartz 2019, Pflieger et al. 2014). Por lo tanto, la entropía de Shannon puede utilizarse para determinar la entropía de un cristal, con la ventaja de poder utilizar una distribución de probabilidades en lugar de números de ocupación de microestados, lo cual se ha propuesto con modelos de interacción por pares (Mayer y Wallek 2020, Wallek et al. 2016). De acuerdo con lo anterior, una parte fundamental en la determinación de la entropía configuracional es identificar el número de posibles configuraciones que un sistema puede adoptar. Esto podría lograrse aplicando directamente el modelo de Mayer y Wallek (2020). Sin embargo, esta propuesta asume que sólo existen seis configuraciones relevantes alrededor de una molécula central, lo cual es algo arbitrario. Así, resulta necesario desarrollar una metodología que permita identificar esas posibles configuraciones, partiendo de su similitud. Esto implica, por una parte, desarrollar una metodología que describa las posiciones y orientaciones de las moléculas en estructuras cristalinas y, por otra parte, emplear esta información para comparar los diferentes ambientes de empaquetamiento para finalmente indicar aquellas configuraciones que sean similares.

Así, en este proyecto se propone una metodología de análisis configuracional de cristales moleculares, como un primer paso para posteriormente determinar la entropía configuracional. En particular, se propone generar un conjunto vasto de estructuras cristalinas de manera aleatoria, tal que se genere un conjunto amplio de posibles configuraciones. Además, se plantea describir y comparar los diferentes ambientes de empaquetamiento en los cristales moleculares empleando cuaterniones duales, un conjunto de ocho números reales que permite describir de manera práctica las posiciones y orientaciones de cuerpos rígidos (moléculas) en el espacio. Tal que, al comparar las diferentes estructuras cristalinas, se puedan agrupar aquellas con configuraciones similares con métodos de agrupamiento o *clustering*. Por lo tanto, en este trabajo se presenta el desarrollo e implementación computacional de la metodología mencionada, así como la proposición y validación de diferentes medidas de comparación entre estructuras cristalinas moleculares; asimismo, se analiza la viabilidad de identificar estructuras cristalinas con configuraciones similares con este tipo de comparaciones.

En resumen, es de gran importancia el desarrollo de nuevos métodos para la determinación de estructuras cristalinas estables de sistemas polimórficos y la inclusión de herramientas que permitan describir la configuración de moléculas en redes cristalinas en este tipo de metodologías puede ser relevante. Por ejemplo, en el cálculo de la entropía de Shannon de manera equivalente a la termodinámica con modelos de interacción por pares, como lo propusieron Mayer et al. (2020, 2016), se requiere de identificar las posibles configuraciones que un mismo sistema cristalino pueda

adoptar. Por lo tanto, esta tesis se enfoca en el desarrollo de una metodología que permita determinar configuraciones cristalinas similares, tal que sea factible asociar esta información con la entropía de los sistemas. Sin embargo, cabe aclarar que la determinación de la entropía configuracional se encuentra fuera del alcance de esta tesis y, por lo tanto, solo se menciona con el fin de establecer el contexto de este proyecto.

2. Marco teórico

2.1. Estudios computacionales del polimorfismo

En esta sección se presenta un breve resumen de algunos trabajos considerados relevantes para el tema de esta tesis. Estos están enfocados en el estudio computacional del polimorfismo y varios de ellos han empleado la entropía como factor importante en sus modelos, lo cual permite contextualizar los objetivos de este proyecto.

El estudio teórico-computacional de fases condensadas ha sido un reto debido a la gran cantidad de interacciones que se pueden dar en estos sistemas. Entre los métodos más socorridos para tratar estos problemas se encuentran los de dinámica molecular, los cuales han sido ampliamente implementados en líquidos. Sin embargo, en el caso de sólidos se requieren de tiempos de cómputo enormes para poder observar el fenómeno de cristalización. Por lo tanto, se han buscado modelos que permitan obtener información de manera más simple.

Uno de los enfoques usuales al momento de trabajar con sólidos son los modelos cuasi-químicos, denominados así dado que se utilizan ecuaciones de acción de masas similares a las utilizadas en el caso de reacciones químicas. Estos modelos consideran la interacción entre partículas por pares y un arreglo no aleatorio de los átomos (Bhadeshia s.f.), aplicando métodos basados en la mecánica estadística. Sin embargo, estos procedimientos suelen introducir grandes errores en las geometrías. Por lo tanto, existen modelos que se basan en cúmulos para ir más allá de las interacciones por pares. Entre estos métodos se destacan los de variación de cúmulos (CVM por sus siglas en inglés) (Kikuchi 1951, Sanchez et al. 1984). Sin embargo, las variables a considerar en estos últimos métodos pueden ser demasiadas y se generan problemas más complejos.

Con el fin de mejorar el modelo anterior, Vinograd y Perchuk (1996) implementaron el concepto de construcción secuencial de una red cristalina como una cadena de Markov. Este modelo toma en consideración el efecto de la geometría sobre las interacciones, lo cual fue una mejora substancial respecto a los métodos previamente descritos. Posteriormente, Mayer y Wallek (2020) extendieron este modelo a sistemas tridimensionales considerando cada punto de la red cristalina como un dado. Así, en este modelo, cada punto de la red tiene seis orientaciones posibles y las interacciones con los puntos vecinos quedan determinadas por pares (interacciones entre caras de los dados), las cuales son caracterizadas por sus energías. Entonces, se toma en consideración tanto la orientación espacial como la energética de las interaccio-

nes. Además, este modelo, al igual que el de Vinograd y Perchuk (1996), se basa en utilizar la entropía de Shannon de manera equivalente a la entropía termodinámica, lo cual facilita definir la entropía del sistema.

El modelo de Mayer y Wallek (2020) lleva a pensar que una molécula que puede estar compuesta por varios grupos funcionales puede tener un número de caras diferente de seis que lleve a interacciones relevantes. Además, este modelo considera que las interacciones en las redes cristalinas se dan siempre entre una molécula central y tres moléculas vecinas; sin embargo, puede haber interacciones importantes con más de tres moléculas cercanas. Por lo tanto, una de las propuestas e hipótesis de este trabajo es que definir *a priori* las diferentes configuraciones que un sistema puede adoptar facilitaría identificar las interacciones relevantes.

Por otra parte, en la actualidad los estudios computacionales para la predicción de cristales moleculares que presentan polimorfismo emplean en su mayoría métodos de aprendizaje automático (o *machine learning*, en inglés). Este término engloba diferentes metodologías, cuyo objetivo principal es la predicción de nuevos datos a partir de datos conocidos. Así, este tipo de métodos han sido ampliamente usados para predecir estructuras cristalinas estables o metaestables, la formación de cocris-tales, y las propiedades fisicoquímicas de nuevos materiales (Heng et al. 2021). En particular, en la predicción de estructuras cristalinas, han habido principalmente dos enfoques: (1) entrenar algoritmos de aprendizaje automático con datos experimentales de un gran conjunto de estructuras cristalinas, los cuales al final predigan nuevas estructuras cristalinas a partir de la estructura molecular, únicamente; (2) generar algoritmos y metodologías que combinen cálculos de estructura electrónica, principalmente con teoría de funcionales de la densidad (*DFT*), y métodos de aprendizaje automático para predecir las nuevas estructuras. Este último tipo de análisis será el empleado en este trabajo.

En la predicción de estructuras cristalinas, uno de los métodos de aprendizaje automático más usados es el de agrupamiento (o *clustering*, en inglés), el cual consiste en dividir los datos en grupos en función de una medida de similitud. Esto permite identificar con mayor facilidad patrones o categorías a partir de datos que pueden estar descritos por una gran cantidad de variables. Esta metodología *per se* no predice nuevos datos, como sí lo hacen otros métodos de aprendizaje automático; sin embargo, esta es útil como un filtro de información de tal manera que puede usarse para delimitar las estructuras cristalinas relevantes de un gran conjunto de estas. Esta idea ha sido central en trabajos como los de AlQuraishi (2019), Piaggi y Parri-nello (2018), Li et al. (2018). Las particularidades de los métodos de agrupamiento se abordarán más adelante.

Por último, resulta importante mencionar que pueden existir una gran cantidad de aproximaciones en los modelos para la predicción de estructuras cristalinas. Por ejemplo, estos usualmente parten de considerar moléculas rígidas; esto es, se considera que la molécula en fase gaseosa mantiene su conformación en la fase cristalina. Asimismo, se asume que la geometría de las moléculas no se modifica por la posición que ocupen en las redes cristalinas; es decir, se desprecia el efecto de las interacciones intermoleculares sobre las geometrías. Por otra parte, también se suele considerar que la estructura más estable a 0 K lo seguirá siendo a otras temperaturas; por lo tanto, también se desprecian los efectos entrópicos (Price 2004). Esta última aproximación puede llevar muchas veces a resultados erróneos; en particular en la predicción de polimorfos ya que la diferencia energética entre estos a 0 K es pequeña, tal que los efectos entrópicos pueden ser determinantes (Nyman y Day 2015). Así, el principal enfoque de este trabajo es establecer una metodología que pueda servir como base para incluir la entropía en el cálculo de la energía libre de cristales moleculares, lo cual a su vez permita mejorar la predicción de sus estructuras cristalinas.

En el caso de sólidos la diferencia en volumen y presión entre dos estructuras cristalinas del mismo material es despreciable (Bernstein 2002). Por lo tanto, la energía libre de estos sistemas depende solamente de la energía interna, ΔE , de la entropía, ΔS , y de la temperatura, T ; lo cual se puede analizar a partir de la energía libre de Gibbs, ΔG :

$$\Delta G = \Delta H - T\Delta S \approx \Delta E - T\Delta S \quad (1)$$

En la ecuación anterior la entalpía se considera aproximadamente igual a la energía interna puesto que, como se mencionó previamente, el cambio en el volumen es despreciable: $\Delta H = \Delta E + P\Delta V$. Además, dado que las posiciones de las moléculas, en los sólidos, están restringidas en las redes cristalinas, se puede asociar a la energía interna con la energía electrónica de manera directa. Así, la energía interna de un sólido puede obtenerse con cálculos de estructura electrónica, lo cual está ampliamente establecido en la literatura. Sin embargo, para calcular la entropía es necesario proponer un modelo. Aunque, considerando igualmente la restricción en las posiciones de las moléculas, la entropía puede aproximarse con solo su componente configuracional. Sin embargo, como se ha mencionado anteriormente, esta última propiedad depende del conocimiento previo de las posibles configuraciones que los sistemas puedan adoptar. Por lo tanto, en las siguientes secciones se abordará, por una parte, la determinación de la energía de estructuras cristalinas con métodos de estructura electrónica y, por otra parte, la descripción de las herramientas ne-

cesarias para la identificación (clasificación) de configuraciones cristalinas similares. Esto último se plantea como una posible estimación de los estados que los sistemas pueden adoptar para el cálculo de la entropía configuracional. Así, se mantendrá en perspectiva la determinación de la energía libre para la predicción de estructuras cristalinas, pero esto no se abordará de manera directa.

2.2. Métodos de estructura electrónica

Una de las partes fundamentales del estudio del polimorfismo de manera teórica es la determinación de la energía total de los sistemas cristalinos. Así, es necesario emplear métodos de estructura electrónica para determinar la energía de red de estos sistemas de manera precisa. De manera general, estas metodologías buscan resolver la ecuación de Schrödinger independiente del tiempo para sistemas multielectrónicos. Para resolver esta ecuación existen diferentes metodologías, como lo son la de Hartree-Fock (*HF*), de interacción de configuraciones (*CI*, por sus siglas en inglés) y de clústeres acoplados (*CC*). Las últimas dos metodologías son altamente precisas; sin embargo, son computacionalmente muy costosas y, por lo tanto, inviables para sistemas de muchos electrones, como suele ser el caso de los sólidos moleculares. Así, estas metodologías no se emplearon en este trabajo; sin embargo, se mencionan dada su relevancia en la química computacional. Además, el método *HF* constituye la base de los métodos actuales de estructura electrónica y, por lo tanto, será el punto de partida en la descripción presentada en esta sección. Asimismo, una alternativa a los métodos de función de onda son los basados en la teoría de funcionales de la densidad (*DFT*). Esta busca, como el nombre lo indica, emplear la densidad electrónica, ρ , en lugar de la función de onda. Este último tipo de método ha sido ampliamente usado en el estudio de sólidos, siendo básicamente la herramienta principal en el estudio teórico-computacional de cristales (Hasnip et al. 2014). Así, esta sección estará centrada en la descripción de la teoría de funcionales de la densidad y su aplicabilidad en el caso de sistemas periódicos.

2.2.1. Métodos multielectrónicos y Hartree-Fock

De acuerdo con el principio de incertidumbre, el cual establece que en una escala subatómica es imposible conocer la posición y el momento de una cierta partícula simultáneamente, los electrones (que son partículas con la misma masa y carga) son indistinguibles. Además, los electrones no son del todo partículas sino que tienen comportamiento ondulatorio. Entonces, la mejor manera de describir sistemas conformados por electrones es considerando una función probabilística. Esto último fue

formalizado por Schrödinger. En particular, en química cuántica se utiliza la ecuación de Schrödinger independiente del tiempo (Ec. 2), ya que la densidad electrónica que define a los sistemas es independiente de este.

$$\hat{\mathcal{H}}\psi = E\psi \quad (2)$$

En la ecuación anterior, $\hat{\mathcal{H}}$ es el operador hamiltoniano, ψ es la función de onda que describe al sistema y E es la energía total del mismo. El hamiltoniano ($\hat{\mathcal{H}}$) genera la descripción energética del sistema, tal que contiene los operadores de energía cinética y potencial (ec. 3). La ecuación 2 tiene solución analítica para el caso de un electrón. Sin embargo, en el caso de sistemas multielectrónicos esta ecuación no tiene solución exacta debido a la interacción entre los electrones.

$$\hat{\mathcal{H}} = - \sum_i^N \frac{1}{2} \nabla_i^2 - \sum_i^N \frac{Z_A}{r_i} + \sum_{i < j}^N \sum_j^N \frac{1}{r_{ij}} \quad (3)$$

En el caso molecular (sistema multielectrónico), el hamiltoniano se expresa como se muestra en la ecuación anterior considerando la aproximación de Born-Oppenheimer (en unidades atómicas). Esta permite separar el movimiento de los núcleos del de los electrones, tal que el operador anterior queda definido para una posición particular de los núcleos (Szabo y Ostlund 1989, pág. 43). En la ecuación 3 se puede ver que el último término depende de la posición de dos electrones, lo cual vuelve al problema inseparable. Es decir, que no se pueda resolver como un conjunto de ecuaciones dependientes de un único electrón. Entonces, una manera alternativa de resolver el problema es proponer una función de onda aproximada, Φ , como productos de funciones monoelectrónicas, ϕ , también denominadas espín-orbitales. Esto último considera el principio variacional, el cual establece que la energía predicha con las funciones aproximadas siempre será mayor o igual a la energía exacta del sistema.

Debido a la indistinguibilidad de los electrones y a que estos siguen la estadística de Fermi-Dirac, las funciones de onda electrónicas que se propongan deben ser antisimétricas (Szabo y Ostlund 1989, pág. 45). Esto se logra representando a la función de onda aproximada de un sistema de N partículas, Φ , como un solo determinante de Slater: el determinante de una matriz con vectores fila asociados a cada partícula y vectores columna asociados a cada espín-orbital o estado. De esta manera se asegura que la función de onda cumple con la condición previamente mencionada. Por otra parte, puesto que no es posible determinar de manera exacta las coordenadas en el espacio tridimensional (posición) de los electrones, se introdujo la coordenada de espín, la cual es una propiedad no-clásica e intrínseca de los electrones. Así, las

funciones propuestas dependen también de la coordenada de espín. La componente magnética del espín tiene dos posibles orientaciones, por lo tanto cada espín-orbital (estado) es ocupado por dos electrones, de acuerdo con el principio de exclusión de Pauli. Por simplicidad se denotará con $\vec{x}$ al conjunto de coordenadas físicas más la coordenada de espín de cada electrón. Entonces, siendo ϕ un espín-orbital y N el número total de electrones, la función de onda se representa de la siguiente manera:

$$\Phi = |\phi_i(\vec{x}_1), \phi_j(\vec{x}_2), \dots, \phi_N(\vec{x}_N)\rangle \quad (4)$$

Con base en lo anterior, el método de Hartree-Fock (*HF*) consiste en proponer una función de onda atómica o molecular, descrita a partir de un conjunto de funciones de base, y resolver el problema por medio del método variacional. Este es un caso particular del método de potencial autoconsistente (*SCF*, por sus siglas en inglés). Así, en el modelo de *HF* las interacciones entre electrones se describen de manera monoeléctronica. En lugar de describir la interacción entre cada par de electrones se describe el potencial que siente cada electrón debido a los demás (Szabo y Ostlund 1989, Cramer 2004). Así, se definen el operador de Coulomb (Ec. 5) y el operador de intercambio (Ec. 6) para cada electrón.

$$\hat{\mathcal{J}}_b(\vec{x}_j)\phi_a(\vec{x}_i) = \left[\int \phi_b(\vec{x}_j) \frac{1}{r_{ij}} \phi_b^*(\vec{x}_j) d\vec{x}_j \right] \phi_a(\vec{x}_i) \quad (5)$$

$$\hat{\mathcal{K}}_b(\vec{x}_j)\phi_a(\vec{x}_i) = \left[\int \phi_b^*(\vec{x}_j) \frac{1}{r_{ij}} \phi_a(\vec{x}_j) d\vec{x}_j \right] \phi_b(\vec{x}_i) \quad (6)$$

En consecuencia, el operador de Fock se define considerando los operadores anteriores y el hamiltoniano monoeléctronico ($\hat{h}_i$), que contiene a los operadores de energía cinética y de interacción núcleo-electrón, de la siguiente manera:

$$\hat{f}(i) = \hat{h}(i) + \sum_{b=1}^N \hat{\mathcal{J}}_b(\vec{x}_j) - \hat{\mathcal{K}}_b(\vec{x}_j) \quad (7)$$

$$\hat{f}(i)\phi_a(\vec{x}_i) = \varepsilon_a \phi_a(\vec{x}_i) \quad (8)$$

Entonces, se tiene una eigen-ecuación por cada electrón y el problema se puede resolver de manera separada. Por otra parte, se puede ver que los operadores de intercambio y correlación dependen a su vez de las funciones sobre las que operan. Por lo tanto, cada vez que se resuelven estas ecuaciones, aplicando el método variacional, se tiene un nuevo conjunto de funciones que se usa para resolver de nuevo el conjunto de ecuaciones. Así, esto es un método iterativo que se repite hasta alcanzar

la convergencia; es decir, hasta que la energía del sistema no cambie dentro de un cierto intervalo de error (Szabo y Ostlund 1989).

Cabe mencionar que en el caso del método *CI* la función de onda se define como una combinación lineal de determinantes de Slater, a diferencia del método *HF* que solo emplea un determinante. Esto implica que se consideran diferentes configuraciones electrónicas de manera simultánea, cada una con diferentes pesos estadísticos; es decir, en la expansión *CI* los coeficientes de cada determinante representan una mezcla estadística de estados (Szabo y Ostlund 1989, pág. 60). Esto se conoce como un método *no local* ya que los electrones se encuentran simultáneamente en orbitales de distintas configuraciones, lo cual a su vez permite mejorar la descripción de la correlación electrónica (interacción entre electrones). En consecuencia, se generan un gran número de integrales bielectrónicas, lo cual lo hace un método computacionalmente muy costoso, tal que solo es factible emplearlo en el caso de sistemas con pocos electrones. Aun así, con *CI*, si se incluyen todas las configuraciones posibles se obtiene la solución exacta del problema de varios electrones, aunque esto es computacionalmente imposible. Por otra parte, el método *CC* parte de la misma idea, pero la expansión de la función de onda se da de manera exponencial y no lineal. Además, se utilizan operadores de clúster en las exponenciales para generar las diferentes configuraciones, lo cual permite optimizar el cálculo de estructura electrónica (Cramer 2004, pág. 224). Así, aunque estos métodos llevan a los resultados más precisos, no es factible usarlos para sistemas con muchos electrones o para analizar un gran conjunto de sistemas, como es el caso en este proyecto.

2.2.2. Funcionales de la densidad

Por otra parte, los métodos de teoría de funcionales de la densidad (*DFT*) están fundamentados en los teoremas de Hohenberg-Kohn. El primero de estos establece que la densidad electrónica determina el potencial externo (debido a los núcleos). Además, se postula que la densidad electrónica determina el número de electrones dada la condición de normalización de la ecuación 9 (Geerlings et al. 2003).

$$\int \rho(\vec{r}) d\vec{r} = N \quad (9)$$

Entonces, como el hamiltoniano depende del potencial externo, este también dependerá de la densidad electrónica. Además, el hamiltoniano determina la energía del sistema. Por lo tanto, esta última también se vuelve dependiente de ρ . Es decir, la energía se vuelve un funcional de la densidad, al igual que sus componentes (Ec. 10). Entonces, se tiene una energía que ya no depende directamente de la determinación

de la función de onda.

$$E[\rho] = E_\nu[\rho] = E_{Ne}[\rho] + T[\rho] + E_{ee}[\rho] \quad (10)$$

El primer término del lado derecho del funcional de la energía representa la interacción núcleo-electrón, el segundo la energía cinética y el último la interacción electrón-electrón. El primer término se puede determinar como en *HF*, puesto que depende de un solo electrón. El último término puede separarse en dos componentes: la interacción coulombica, $J[\rho]$, y la de intercambio y correlación E_{xc} (Ec. 11).

$$E_{ee}[\rho] = J[\rho] + E'_{xc}[\rho] \quad (11)$$

El funcional de interacción coulombica (Ec. 12) se conoce ya que cada electrón ocupa un solo orbital y la densidad de un orbital no es más que el producto de la función de onda por su conjugado. Sin embargo, en el caso del funcional de intercambio y correlación ($E'_{xc}[\rho]$) no se conoce su expresión exacta, ya que involucra la interacción entre electrones que ocupan dos orbitales de manera simultánea, tal que no es posible expresarla en términos de la densidad electrónica de manera directa. Por lo tanto, este último término, E'_{xc} , es una de las principales incógnitas en este modelo. La otra incógnita es el funcional de la energía cinética, $T[\rho]$ (Perdew y Kurth 2003).

$$J[\rho] = \int \int \frac{\rho(\vec{r}_1)\rho(\vec{r}_2)}{r_{12}} d\vec{r}_1 d\vec{r}_2 \quad (12)$$

Entonces, se desconoce la forma de $T[\rho]$ y $E'_{xc}[\rho]$. Una de las primeras propuestas para resolver esto fue hecha por Kohn y Sham (1965). Su aproximación parte de considerar un sistema de partículas independientes de referencia con la misma densidad que el sistema real, obtener su energía cinética exacta (T_S) e incluirla en el funcional general (Ec. 13). Para tomar en cuenta que el sistema no es realmente de partículas independientes, Kohn y Sham introdujeron todas las expresiones desconocidas en el funcional de intercambio y correlación (Ec. 14), el cual difiere del mostrado anteriormente para el funcional exacto bajo el esquema de Hohenberg y Kohn (ecs. 10 y 11).

$$F[\rho] = T_S[\rho] + J[\rho] + E_{XC}[\rho] \quad (13)$$

$$E_{XC} = (T[\rho] - T_S[\rho]) + (E_{ee}[\rho] - J[\rho]) \quad (14)$$

Así, dado que este método también considera la aproximación de partículas independientes, el método de Kohn y Sham también emplea espín-orbitales mono-electrónico, lo cual a su vez lleva a las siguientes ecuaciones:

$$\left(-\frac{1}{2}\nabla_{\vec{r}}^2 + \int d\vec{r}' \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} + \frac{\delta E_{xc}[\rho]}{\delta \rho(\vec{r})} + \nu(\vec{r}) \right) \phi_i(\vec{x}) = \epsilon_i \phi_i(\vec{x}) \quad (15)$$

donde ν es el potencial externo debido a los núcleos y

$$\rho(\vec{r}) = \sum_{b=1}^N |\phi_b(\vec{r})|^2 \quad (16)$$

De este modo, esta metodología también funciona con base en el principio variacional. Es decir, se busca el conjunto de orbitales de Kohn y Sham que minimicen la energía y que cumplan con las condiciones de frontera (Ec. 17). Sin embargo, esto evidentemente depende de que se conozca la forma del funcional de intercambio-correlación, E_{xc} .

$$E_0 = \min_{\rho \rightarrow N} (F[\rho] + \int \rho(\vec{r}) V_{Ne} d\vec{r}) \quad (17)$$

Existe una gran variedad de funcionales de la densidad con diferentes descripciones y aproximaciones de la energía de intercambio-correlación. Las aproximaciones permiten acelerar los tiempos de cálculo, de otra manera serían tan costosos como los métodos de función de onda. Sin embargo, muchas de estas son definidas para aplicaciones específicas. Por lo tanto, no existe una formulación del funcional de intercambio-correlación general. La mejor manera de seleccionar uno es buscar el que mejor describa el tipo de sistemas con los que se quiere trabajar (Rappoport et al. 2009). Así, en las siguientes secciones se mostrarán las bases de uno de los funcionales de intercambio-correlación más empleados en sólidos: *PBE* (Perdew et al. 1996).

Métodos de gradiente generalizado (GGA)

La aproximación *base* de los funcionales de intercambio consiste en considerar un funcional que depende únicamente de la densidad electrónica del sistema y que además esta última (ρ) es homogénea, lo cual se conoce como aproximación de densidad local (*LDA*, por sus siglas en inglés). Esta aproximación surge de considerar la densidad de un gas homogéneo de electrones, tal que se obtiene lo siguiente (Parr y Weitao 1995):

$$E_x^{LDA}[\rho] = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \int \rho(\vec{r})^{4/3} d\vec{r} \quad (18)$$

La principal desventaja de la *LDA* es justamente considerar que la densidad electrónica es homogénea, lo cual solo en pocos casos es válido. Por lo tanto, una corrección importante al funcional anterior es agregar la dependencia en el gradiente de la densidad. De manera general, esto se logra al agregar una función de exacerbamiento para considerar la inhomogeneidad de la densidad electrónica, lo cual se conoce como métodos de gradiente generalizado (GGA):

$$E_x[\rho] = \int \rho(\vec{r})^{4/3} F_x^{GGA}(s) d\vec{r} \quad (19)$$

donde s es el gradiente reducido de la densidad:

$$s(\vec{r}) = \frac{|\nabla \rho(\vec{r})|}{2(3\pi^2)^{1/3} \rho(\vec{r})^{4/3}} \quad (20)$$

y F_x^{GGA} es la función de exacerbamiento, cuya definición varía y da origen a una gran variedad de aproximaciones del funcional de intercambio. Cabe mencionar que esta no es la única corrección empleada, por ejemplo también existen funcionales que emplean el laplaciano de la densidad, conocidos como meta-GGA. Sin embargo, el funcional empleado en este trabajo es del tipo GGA y, por lo tanto, se mantendrá la discusión hasta este punto.

En particular, uno de los funcionales más empleados en sólidos es el de Perdew, Burke y Ernzerhof (*PBE*) (Rappoport et al. 2009, Hasnip et al. 2014), el cual usa la definición del factor de exacerbamiento propuesta por Perdew et al. (1996),

$$F_x^{PBE}(s) = 1 + \kappa - \frac{\kappa}{1 + \mu s^2 / \kappa} \quad (21)$$

donde κ y μ son parámetros ajustados con condiciones que cumple el funcional exacto. Así, una de las ventajas es que este funcional ha sido parametrizado con condiciones exactas y no con datos experimentales o con casos particulares. Asimismo, se tiene para el funcional de correlación:

$$F_c^{PBE}(\rho_\downarrow, \rho_\uparrow) = \int \rho(\vec{r}) [\epsilon_c^{LDA}(r_s, \zeta) + H(r_s, \zeta, t)] d\vec{r} \quad (22)$$

donde $\rho_\downarrow$ y $\rho_\uparrow$ son las componentes de espín de la densidad electrónica, tal que $\rho(\vec{r}) = \rho_\downarrow(\vec{r}) + \rho_\uparrow(\vec{r})$; la función H de igual manera se formula siguiendo el comportamiento del funcional exacto.

En general, el funcional *PBE* es ampliamente usado en el estudio de sistemas cristalinos. En particular, en el caso de cristales moleculares, se ha demostrado que permite obtener energías reticulares de manera bastante precisa y ha resultado útil en la predicción de polimorfos (Brandenburg y Grimme 2016). De acuerdo con el artículo de Brandenburg y Grimme (2016), *PBE* no es el funcional que mejor se desempeña en la predicción de polimorfos, al menos en el trabajo mencionado; sin embargo, su desempeño sigue siendo adecuado. Por otra parte, es importante remarcar que la precisión en los cálculos de energía con el funcional *PBE* depende de añadir correcciones de interacciones de dispersión de London (Brandenburg y Grimme 2016). Por lo tanto, el funcional *PBE* se implementó incluyendo la corrección de dispersión de Tkatchenko-Scheffler (TS), la cual se describe en la siguiente sección.

Corrección de dispersión

Una de las principales desventajas tanto en Hartree-Fock como en *DFT* es que estos métodos fallan en describir ciertas interacciones intermoleculares. Esto es, interacciones de largo alcance que son generalmente conocidas como fuerzas de dispersión de London. Este tipo de interacciones dependen principalmente de una descripción correcta de la correlación electrónica. Por lo tanto, *HF* falla en este caso al incluir solo una parte de la correlación al ser un método monodeterminantal (aproximación de partículas independientes). En el caso de *DFT*, esto último también ocurre y, además, las aproximaciones al funcional de intercambio-correlación suelen ser funcionales de la densidad local así como de su gradiente (en el caso de GGA) (Foster y Sohlberg 2010). Por lo tanto, es necesario incorporar correcciones tal que pueda incluirse el efecto de las interacciones intermoleculares. Una de las metodologías usuales para incluir esta corrección es la propuesta por Tkatchenko y Scheffler (Bučko et al. 2013), la cual propone una contribución a la energía por fuerzas de dispersión de la siguiente forma:

$$E_{disp} = -\frac{1}{2} \sum_A^N \sum_B^N \sum_L^I \frac{C_{6AB}}{(r^{AB,L})^6} f_{damp}(r^{AB,L}) \quad (23)$$

donde $r^{AB,L} = |r^{A,0} - r^{B,L}|$, las sumas son sobre los índices de los N átomos y todas las traslaciones de la celda unitaria $L = (l_1, l_2, l_3)$, el término primado indica que $A \neq B$ si $L=0$ (Bučko et al. 2013). Además, f_{damp} es una función que permite asegurar que las interacciones consideradas sean interatómicas al eliminar aquellas que se den a distancias muy cortas:

$$f_{damp}(r^{AB,L}) = \frac{1}{1 + \exp[-d(\frac{r^{AB,L}}{s_R R_{AB}^{eff}}) - 1]} \quad (24)$$

donde d es un parámetro fijo, s_R es un parámetro ajustado a cada método *DFT* y R_{AB}^{eff} es la suma de los radios de van der Waals de cada átomo. Asimismo, los coeficientes C en la energía de dispersión (ec. 23) dependen de los volúmenes efectivos atómicos (ν_i) y su relación respecto a los mismos coeficientes medidos en átomos libres:

$$\alpha_i = \nu_i \alpha_i^{libre} \quad (25)$$

$$C_{6ii} = \nu_i^2 C_{6ii}^{libre} \quad (26)$$

donde ν_i se define de acuerdo con la partición de Hirshfeld (Hirshfeld 1977):

$$\nu_i = \frac{\int r^3 \omega_i(\vec{r}) \rho(\vec{r}) d\vec{r}}{\int r^3 \rho_i^{libre}(\vec{r}) d\vec{r}} \quad (27)$$

$$\omega_i(\vec{r}) = \frac{\rho_i^{libre}(\vec{r})}{\sum_{j=1}^{N_{at}} \rho_j^{libre}(\vec{r})} \quad (28)$$

Así, el método Tkatchenko-Sheffler (*TS*) permite incluir las interacciones por dispersión en cálculos de estructura electrónica. Esta metodología es principalmente importante en el caso del análisis de cristales moleculares dado que este tipo de interacciones son fundamentales. La definición aquí presentada es la implementada en el programa VASP y fue utilizada para todos los cálculos de energía realizados.

2.2.3. Sistemas periódicos

Por otra parte, se describe en esta sección cómo los métodos de estructura electrónica se pueden aplicar a redes cristalinas. Los cristales se caracterizan por una unidad estructural mínima, conocida como celda unitaria, la cual se repite de manera periódica en el espacio generando todo el cristal. Así, todos los átomos, moléculas o iones del cristal, asociados a puntos de la red cristalina, pueden generarse a partir de trasladar la celda unitaria con un vector $\vec{R}$, de acuerdo con la siguiente ecuación:

$$\vec{R} = n_1 \vec{a}_1 + n_2 \vec{a}_2 + n_3 \vec{a}_3 \quad \text{con} \quad n_1, n_2, n_3 \in \mathbb{Z} \quad (29)$$

donde $\vec{a}_1$, $\vec{a}_2$ y $\vec{a}_3$ son los vectores que describen a la celda unitaria. Entonces, dado que las posiciones de los núcleos se repiten por cada desplazamiento en $\vec{R}$, el potencial generado por estos también debe mantener esta periodicidad. En otras palabras, la principal característica de los sistemas cristalinos es que el potencial que siente cada uno de los electrones es periódico:

$$V^{sp}(\vec{r} + \vec{R}) = V^{sp}(\vec{r}) \quad (30)$$

Esto, de acuerdo con el teorema de Bloch, resulta en que las funciones de onda de estos sistemas también deben ser periódicas ante la traslación considerando un cierto factor de fase:

$$\psi_k(\vec{r} + \vec{R}) = e^{i\vec{k} \cdot \vec{R}} \psi_k(\vec{r}) \quad (31)$$

donde $\vec{k}$ puede entenderse como un vector de onda, conocido como momento del cristal. Esto implica que la función de onda no decae de una celda unitaria a otra, sino que se extiende por todo el cristal (Kaxiras 2003, Cap. 3). Así, este término permite tomar en cuenta diferentes posibles fases que los electrones pueden tener de una celda unitaria a otra.

Así, dada la periodicidad de la función de onda, basta con conocer su valor dentro de la celda unitaria. Asimismo, resulta conveniente trabajar con su equivalente en un espacio recíproco, conocida como primera zona de Brillouin (BZ , por sus siglas en inglés). El espacio recíproco se define tal que los vectores que describen a la primera zona de Brillouin, $\vec{b}_j$, cumplen con la siguiente relación con los vectores equivalentes en el espacio real: $\vec{a}_i \cdot \vec{b}_j = 2\pi\delta_{ij}$. Entonces, se tiene un vector $\vec{G}$ correspondiente a $\vec{R}$ que conecta todos los puntos equivalentes en la red:

$$\vec{G} = m_1\vec{b}_1 + m_2\vec{b}_2 + m_3\vec{b}_3 \quad \text{con} \quad m_1, m_2, m_3 \in \mathbb{Z} \quad (32)$$

Tal que,

$$\vec{G} \cdot \vec{R} = 2\pi l, \quad l = n_1m_1 + n_2m_2 + n_3m_3 \quad (33)$$

$$e^{i\vec{G} \cdot \vec{R}} = 1 \quad (34)$$

De acuerdo con lo anterior, una forma alternativa de definir el teorema de Bloch, es la siguiente:

$$\psi_{\vec{k}}(\vec{r}) = e^{i\vec{k}\cdot\vec{r}} u_{\vec{k}}(\vec{r}) \quad (35)$$

$$u_{\vec{k}}(\vec{r}) = \sum_{\vec{G}} \alpha_{\vec{k}}(\vec{G}) e^{i\vec{G}\cdot\vec{r}} \quad (36)$$

En consecuencia, se sigue cumpliendo que la función de onda es periódica, ya que $u_{\vec{k}}(\vec{r} + \vec{R}) = u_{\vec{k}}(\vec{r})$. Además, esta última función se expresa como una combinación lineal de ondas planas ($e^{i\vec{G}\cdot\vec{r}}$). Así, al aplicar el hamiltoniano para una sola partícula, se obtiene lo siguiente (Kaxiras 2003, pág. 16):

$$\sum_{\vec{G}} \left[\frac{1}{2}(\vec{k} + \vec{G})^2 - \epsilon_{\vec{k}} + \sum_{\vec{G}'} V(\vec{G}') e^{i\vec{G}'\cdot\vec{r}} \right] \alpha_{\vec{k}}(\vec{G}) e^{i(\vec{k}+\vec{G})\cdot\vec{r}} = 0 \quad (37)$$

Entonces, la energía de los sistemas periódicos depende del número cuántico $\vec{k}$. La ecuación anterior se resuelve de acuerdo con el principio variacional, tal que se buscan los coeficientes $\alpha_{\vec{k}}(\vec{G})$ que minimizan la energía. La resolución de la ecuación anterior se abordará en la siguiente sección desde la perspectiva del método del proyector de ondas aumentadas (*PAW*, por sus siglas en inglés).

Por otra parte, cabe mencionar que existe una infinidad de ondas planas (ec. 32) que pueden considerarse al resolver la ecuación anterior (ec. 37). Por lo tanto, en la práctica es necesario acotar el número de funciones consideradas, lo cual se indica como un corte en energía. Esto se debe a que se puede obtener la energía cinética asociada a una onda plana:

$$\langle e^{i(\vec{k}+\vec{G})\cdot\vec{r}} | -\frac{\hbar}{2m} \nabla_{\vec{r}}^2 | e^{i(\vec{k}+\vec{G})\cdot\vec{r}} \rangle = \frac{\hbar^2}{2m} |\vec{k} + \vec{G}|^2 \quad (38)$$

Así, se asigna un valor de energía E_{cut} , tal que solo se usan ondas planas cuyas energías cinéticas son menores a esta, para cada punto $\vec{k}$. Evidentemente, la elección de este parámetro depende de cada sistema, por lo que es necesario asegurar la convergencia de la energía respecto al valor de E_{cut} .

Además, como se mostró en las secciones anteriores, para el cálculo de la energía de cada estado es necesario integrar sobre todo el espacio. Asimismo, en este caso, se tienen que calcular las integrales sobre el espacio de $\vec{k}$. Sin embargo, estas integrales pueden aproximarse con un conjunto finito de puntos $\vec{k}$ particulares, tal que la integral de cualquier función periódica F puede aproximarse de la siguiente manera (Kratzer y Neugebauer 2019):

$$\frac{\Omega}{(2\pi)^3} \int_{BZ} F(\vec{k}) d\vec{k} \approx \sum_{j=1}^M w_j F(\vec{k}_j) \quad (39)$$

donde Ω es el volumen de la celda unitaria en el espacio real, BZ implica que la integral se realiza en la primera zona de Brillouin, M es el número total de puntos $\vec{k}$ considerados y w_j son los pesos asignados a cada punto $\vec{k}$ en función de su simetría. Existen diferentes propuestas para identificar los puntos $\vec{k}$ relevantes (Moreno y Soler 1992, Wisesa et al. 2016, Hart et al. 2019), las cuales en general emplean la simetría de los sistemas cristalinos tal que los puntos $\vec{k}$ sean representativos y, así, la aproximación mostrada en la ecuación 39 sea válida. Aun así, la elección de la cantidad de puntos $\vec{k}$ necesarios se desconoce *a priori*, ya que dependen de cada sistema. Por lo tanto, también es necesario analizar la convergencia de la energía respecto al número de puntos $\vec{k}$ considerados.

Método del proyector de ondas aumentadas (PAW)

Una de las principales desventajas al definir la función de onda como una combinación lineal de ondas planas (ec. 36) es que se requiere de un gran conjunto de estas últimas para lograr representar la función de onda adecuadamente cerca de los núcleos. Esto se debe a que en esta región la función de onda oscila considerablemente. Sin embargo, a partir de una cierta distancia de los núcleos, la función se vuelve suave; es decir, es una función continua que oscila poco. Así, resulta conveniente definir orbitales centrados en los núcleos compuestos de una función suave o de valencia, la cual se puede desarrollar en ondas planas, y de funciones cercanas a los núcleos o de *core*, las cuales pueden obtenerse de la solución completa de la ecuación de Kohn y Sham para los átomos aislados. Este método fue propuesto por Blöchl (1994) y la descripción presentada en esta sección está basada en la desarrollada por Rostgaard (2009).

La función de onda de valencia presenta un comportamiento suave (oscila muy poco). Por lo tanto, si se parte de un conjunto de funciones suaves para aproximar la función de onda, no es necesario transformarlas para que estas reproduzcan el comportamiento real en la región de valencia. Así, solo es necesario incluir una transformación lineal que lleve de las funciones cercanas al núcleo a las reales, puesto que la función de onda en esta región no es suave:

$$\hat{\tau} = 1 + \sum_a \hat{\tau}^a \quad (40)$$

donde $\hat{\tau}^a$ son transformaciones centradas en los núcleos a , tal que solo actúan en la región $|\mathbf{r} - \mathbf{R}^a| < r_c^a$, para un radio de corte, r_c^a , dado.

Así, se tiene la siguiente relación entre funciones auxiliares de espín orbital, $|\tilde{\phi}_i^a\rangle$, y los orbitales de *core*, $|\phi_i^a\rangle$:

$$|\phi_i^a\rangle = (1 + \hat{\tau}^a) |\tilde{\phi}_i^a\rangle \leftrightarrow \hat{\tau}^a |\tilde{\phi}_i^a\rangle = |\phi_i^a\rangle - |\tilde{\phi}_i^a\rangle \quad (41)$$

Esta separación permite a su vez distinguir entre la energía dependiente de las funciones suaves y la energía de *core*:

$$E = \tilde{E} + \sum_a \Delta E_a \quad (42)$$

tal que $\tilde{E}$ es el funcional de energía evaluado en las funciones suaves,

$$\tilde{E} = T_s[\{\tilde{\psi}_n\}] + J[\tilde{\rho}] + E_{XC}[\tilde{\rho}] \quad (43)$$

El conjunto de orbitales $\tilde{\psi}$ pueden desarrollarse en términos de pocas ondas planas, de acuerdo con lo presentado en la sección anterior. Por otra parte, las componentes de *core* dependen de varios tensores,

$$\Delta E^a = \Delta T_c^a + \Delta C^a + \sum_{i_1 i_2} (\Delta T_{i_1 i_2}^a + \Delta C_{i_1 i_2}^a) + \sum_{i_1 i_2 i_3 i_4} D_{i_1 i_2}^{a*} \Delta C_{i_1 i_2 i_3 i_4}^a D_{i_3 i_4}^a + \Delta E_{xc}^a[\{D_{i_1 i_2}^a\}] \quad (44)$$

tal que los coeficientes T_c^a , $T_{i_1 i_2}^a$, $C_{i_1 i_2}^1$, y $\Delta C_{i_1 i_2 i_3 i_4}^a$ son términos que no dependen de los sistemas y, por lo tanto, se han calculado previamente para la mayoría de los átomos en la tabla periódica (Kresse y Joubert 1999a), lo cual permite optimizar los cálculos de energía.

2.3. Métodos de agrupamiento

La agrupación de datos es un tema ampliamente estudiado y explorado a través de los métodos de agrupamiento (*clustering*, en inglés) (Celebi et al. 2013, Jain 2010). Estas metodologías son una rama del aprendizaje automático del tipo no supervisado, ya que estos no dependen de conocer resultados previos. Este tipo de algoritmos permiten condensar la información de un gran conjunto de datos al asociar aquellos que son similares. En química, estos han sido ampliamente usados para la selección de estructuras químicas relevantes a partir de un grupo de estas. En particular, han sido útiles en la predicción de estructuras de proteínas (AlQuraishi 2019) y de cristales moleculares (Piaggi y Parrinello 2018, Li et al. 2018).

Los métodos de agrupamiento se basan en tres principios fundamentales (Xu y Tian 2015):

- Se debe tener una medida de similitud, s , o de disimilitud, d , entre pares de datos.
- Los datos pertenecientes a una misma agrupación o clúster deben tener la mayor similitud posible entre ellos.
- Los datos pertenecientes a diferentes clústeres deben ser lo más diferentes posible.

Así, la agrupación de datos parte de poder medir la similitud o disimilitud entre estos, lo cual dependerá de su propia naturaleza. Sin embargo, de manera general, se debe cumplir que si se tiene una métrica o medida de disimilitud, d , esta debe ser nula al comparar las mismas estructuras ($d_{ii} = 0$), simétrica ($d_{ij} = d_{ji}$) y no-negativa ($d_{ij} \geq 0$). En el caso de una medida de similitud, s , esta debe también cumplir con las últimas dos propiedades, pero esta deberá tomar valores grandes si los datos comparados son similares (Meila 2016). Por ejemplo, si se tienen puntos en un espacio vectorial tridimensional una buena métrica es la distancia euclidiana. Sin embargo, el concepto de métrica es general, tal que este puede implicar una comparación más abstracta o no tan intuitiva. En el caso particular de moléculas y sistemas cristalinos, se han propuesto una gran variedad de métricas (Onat et al. 2020, Langer et al. 2022, Sadeghi et al. 2013), las cuales consideran tanto aspectos estructurales como de reactividad química de los sistemas, lo cual se detallará más adelante, en particular para las medidas empleadas en este trabajo.

Una vez planteado un parámetro de comparación es necesario implementar un algoritmo que permita hacer las agrupaciones de acuerdo con los dos últimos fundamentos de los métodos de *clustering*, lo cual también depende del tipo de datos (Xu

y Tian 2015). Por lo tanto, en esta sección se presentan tres métodos de agrupamiento que se considera pueden ser útiles para el análisis del espacio configuracional de estructuras cristalinas moleculares.

2.3.1. *K-means*

Entre los métodos de agrupamiento más simples y a su vez más empleados se encuentran los denominados como *k-means* (Celebi et al. 2013, Jain 2010). Este tipo de algoritmos se basan directamente en los principios mencionados anteriormente; es decir, minimizar las distancias entre los datos de un mismo clúster y maximizar a su vez aquellas entre los pertenecientes a diferentes clústeres. Esto equivale a minimizar una función de error cuadrático medio, también denominada inercia, como la que se muestra a continuación:

$$W(S, C) = \sum_{k=1}^K \sum_{i \in S_k} |y_i - c_k|^2 \quad (45)$$

donde K es el número de agrupaciones o clústeres, C es el conjunto de centros o centroides de cada clúster (c_k), S es el conjunto de datos analizados, S_k es el subconjunto de datos que conforman a un clúster y y_i es el vector que representa a un dato del subconjunto. La búsqueda del mínimo global de esta función (ec. 45) es computacionalmente imposible, excepto para conjuntos pequeños de datos (Kaufman y Rousseeuw 1990, Lloyd 1982). Por lo tanto, la resolución de este problema se logra empleando algoritmos heurísticos, de acuerdo con los siguientes pasos (Kodinariya y Makwana 2013, Arthur y Vassilvitskii 2007, Lloyd 1982):

1. Se seleccionan k centros (número de clústeres) asignándolos de manera aleatoria a datos del conjunto de entrada, $c_1, \dots, c_k \in S$
2. Asignar cada dato al centro más cercano de acuerdo con la distancia euclidiana entre los vectores correspondientes.
3. Recalcular las posiciones de los k centros como el promedio de los valores asignados a cada centro en el paso anterior.
4. Repetir los pasos 2 y 3 hasta la convergencia. Es decir, hasta que las posiciones de los centros o las asignaciones de los datos a cada grupo no cambien de una iteración a otra, dado un cierto límite, o hasta que el número de iteraciones máximo se alcance.

El método descrito previamente es ampliamente usado y constituye la base de varios métodos de agrupamiento (Xu y Tian 2015). Sin embargo, es un método que falla si la distancia euclidiana no es una medida apropiada para el conjunto de datos, lo cual también implica que los clústeres se alejan de una forma esférica o hiper-esférica (Fleshman 2019). Además, es un método que depende de la descripción de los datos por medio de vectores, lo cual no siempre se tiene. También, este método depende de elegir previamente el número de clústeres a formar y, por lo tanto, se debe recurrir a métodos adicionales para determinar este hiperparámetro (k). Es decir, k es un parámetro que se da al inicio del algoritmo y se mantiene constante durante todas las iteraciones del método, tal que resulta necesario aplicar varias veces el procedimiento *k-means* para optimizar su valor (Kodinariya y Makwana 2013).

De acuerdo con lo anterior, el resultado del método *k-means* depende de la inicialización de los centroides. Así, una primera manera de compensar el efecto de los centros iniciales sobre las agrupaciones finales es ejecutar el algoritmo varias veces y seleccionar los clústeres que minimicen la función de error, W (ec. 45). Sin embargo, esto es computacionalmente costoso y no existe manera de asegurar que con más repeticiones se obtenga un mejor resultado. Por lo tanto, se han propuesto algoritmos para la inicialización de los centroides (Celebi et al. 2013). En particular, se ha mostrado que los algoritmos demoninados como *k-means++* y su variación *greedy k-means++* llevan a buenos resultados (Celebi et al. 2013). El algoritmo *k-means++* fue propuesto por Arthur y Vassilvitskii (2007) e implica los siguientes pasos:

1. Elige a alguno de los datos $y \in S$ como el primer centroide de manera aleatoria:
 $C_1 = \{y\}$
2. Calcula las distancias $d(y, C_i) = \min_{c \in C_i} |y - c|^2$, entre cada punto y el centroide más cercano, previamente elegido.
3. Elige un nuevo punto $y \in S$ como centroide, tal que la probabilidad de elegir a un punto como centroide es $p = \frac{d(y, C_i)}{\sum_i^N d(y, C_i)}$.¹ Se establece $C_{i+1} = C_i \cup \{y\}$
4. Se repite los pasos 2 y 3 hasta obtener k centroides.
5. Se continua con los pasos estándar del algoritmo *k-means*, previamente mencionados.

Cabe mencionar que, en el paso 2 del algoritmo previo, la elección del nuevo centroide puede variar. Algunas implementaciones del algoritmo seleccionan direc-

¹ Así, los puntos más alejados de los centroides previamente elegidos tienen mayor probabilidad de elegirse como nuevo centroide.

tamente el punto con la mayor probabilidad, p , como el nuevo centroide y algunas otras emplean la distribución completa de probabilidades pesadas, p , para elegir de manera aleatoria el centroide. Esta última es la implementada en la librería *scikit-learn* (Pedregosa et al. 2011).

Por otra parte, el algoritmo conocido como *greedy k-means++*, también propuesto por Arthur y Vassilvitskii (2007), implica evaluar múltiples centroides en cada paso:

1. Elige de manera aleatoria l centroides, $c_1^1, \dots, c_l^1 \in X$
2. Se elige el centroide que minimiza la función de error, $c_1 = \operatorname{argmin}_{c \in \{c_1^1, \dots, c_l^1\}} W(X, c)$ y se establece $C_1 = \{c_1\}$.
3. Calcula las distancias $d(x, C_i)$.
4. Elige un nuevo conjunto de centroides $c_{i+1}^1, \dots, c_{i+1}^l \in X$, tal que la probabilidad de elegir a un punto como centroide es $p = \frac{d(x, C_i)}{\sum_i^N d(x, C_i)}$.
5. Se establece $c_{i+1} = \operatorname{argmin}_{c \in \{c_{i+1}^1, \dots, c_{i+1}^l\}} W(X, C_i \cup \{c\})$ y $C_{i+1} = C_i \cup \{c_{i+1}\}$
6. Se repiten los pasos 3 a 5 hasta obtener k centroides.
7. Se continua con los pasos estándar del algoritmo *k-means*, previamente mencionados.

En general, a pesar de que no existe una diferencia considerable entre las agrupaciones obtenidas con ambos algoritmos, el método *greedy k-means++* puede llevar a mejores resultados (Celebi et al. 2013, Grunau et al. 2022). Así, considerando lo anterior se utilizó en este trabajo el método *greedy k-means++*. Además, este algoritmo es el implementado en la paquetería *scikit-learn* (Pedregosa et al. 2011), lo cual facilitó su uso. Sin embargo, este método no se aplicó directamente en este trabajo debido al tipo de datos obtenidos; sino que se utilizó a través de otros métodos conocidos como espectrales, los cuales se describen a continuación.

2.3.2. Métodos espectrales

Una alternativa al método *k-means* surge de la teoría de grafos (von Luxburg 2007). En esta teoría se tiene una representación de datos o información a través de un conjunto de vértices (V) y las aristas que los unen (E). Formalmente un grafo se define como $G = (V, E)$. Estas representaciones pueden asociarse a una matriz de adyacencia en la cual se indica la arista entre cada par de vértices: $\mathbf{W} = (w_{ij})_{i,j=1,\dots,n}$.

En el caso más simple, si existe una conexión entre los vértices v_i y v_j , $w_{ij} = 1$, en el caso contrario $w_{ij} = 0$. De manera general, cada arista puede tener un cierto peso, sea $w_{ij} \geq 0$, y la conexión se puede dar en ambos sentidos ($w_{ij} = w_{ji}$), lo cual se conoce como grafo ponderado no direccionado. Este último tipo de matriz o de grafo es la base de los métodos espectrales.

La matriz de adyacencia se puede construir para cualquier tipo de datos si para estos se tiene alguna medida de similitud, s . Esto es, como se mencionó previamente, una medida no-negativa que toma valores grandes para datos similares y valores pequeños para datos diferentes. Con esta medida se construye una matriz de similitud, $\mathbf{S} = (s_{ij})_{i,j=1,\dots,n}$. A partir de esta última existen diferentes maneras de construir la matriz de adyacencia (Meila 2016, von Luxburg 2007), aunque solo se hará referencia a la implementada en este trabajo. Esto es, se puede considerar a la matriz de similitud, $\mathbf{S}$, equivalente a la matriz de adyacencia, $\mathbf{W}$ (von Luxburg 2007). En otras palabras, se considera que todos los vértices están conectados y el peso de las aristas está dado por la medida de similitud.

Una vez que se genera la matriz de adyacencia, $\mathbf{W}$, a partir de esta se construye la matriz de grado, $\mathbf{D}$. Esta última es diagonal y cada uno de sus elementos (d_i) se compone de la suma de los valores en cada renglón de la matriz de adyacencia (ec. 46). Esta matriz, $\mathbf{D}$, es una medida de la conexión total de cada vértice.

$$d_i = \sum_{j=1}^n w_{i,j} \quad (46)$$

Finalmente, empleando la matriz de grado se construye la matriz $\mathbf{L}$, conocida como la matriz laplaciana normalizada o simplemente el laplaciano (Chung 1997):

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{S} \mathbf{D}^{-1/2} \quad (47)$$

Asimismo, se puede emplear un laplaciano no normalizado u otros tipos de normalización (von Luxburg 2007, Meila 2016). Sin embargo, lo presentado aquí es un método estándar y se ha usado como un primer acercamiento al análisis con este tipo de métodos. Además, en el caso en que $\mathbf{W}$ es un grafo ponderado, lo cual es el caso de las matrices analizadas en este trabajo, la normalización del laplaciano es indispensable para evitar que vértices con un alto grado, d , (alta conectividad) sesguen los resultados (von Luxburg 2007).

De acuerdo con lo anterior, los métodos espectrales consisten en el análisis de la matriz laplaciana. Las propiedades relevantes de esta matriz para el método de agrupamiento son que sus valores propios son positivos y la cantidad de valores

propios iguales a cero equivale al número óptimo de clústeres (Fleshman 2019, von Luxburg 2007). Sin embargo, en la práctica, sin tomar en cuenta el primer eigenvalor, el cual siempre es nulo (todos los datos corresponden a un solo clúster), se pueden obtener varios eigenvalores cercanos a cero, pero no exactamente iguales a cero, ya que suelen existir conexiones entre todos los datos. Así, una manera práctica de determinar el número de clústeres es analizar gráficamente los eigenvalores y seleccionar los más pequeños, cuyos valores deben ser cercanos, el número de estos equivale al número de clústeres. En otras palabras, el análisis de los eigenvalores de la matriz laplaciana permite conocer la cantidad óptima de grupos que se pueden formar con el conjunto de datos analizado.

De acuerdo con el método de Ng et al. (2001), una vez analizados los eigenvalores de la matriz laplaciana, el agrupamiento se realiza a partir de una proyección de los datos en un espacio vectorial de menor dimension. Esto se logra al construir una matriz de tamaño $n \times k$, donde n es el número total de datos y k el número de eigenvalores de L más pequeños. Esta matriz se construye tal que sus columnas están conformadas por los eigenvectores correspondientes a los eigenvalores más pequeños. Este método se conoce como *spectral embedding*, el cual se basa en que los vértices cercanos en el grafo corresponden a vectores cercanos en el espacio vectorial generado (Bonald 2019-2020). Esto último permite implementar el método de *k-means* para finalmente obtener las agrupaciones del conjunto de datos a partir de la matriz construida con los eigenvectores (Fleshman 2019, Ng et al. 2001).

Algunas de las ventajas de los métodos espectrales son que estos no dependen de una distancia euclidiana entre los datos sino de su similitud en sí. Esto a su vez implica que las agrupaciones no asumen una forma en particular. Aunque al final del algoritmo previamente mencionado se aplica el método *k-means*, el cual asume regiones esféricas o hipersféricas, el método de *spectral embedding* permite realizar el mapeo de regiones que no necesariamente son convexas a un espacio en el que estas sí lo son (Ng et al. 2001). Por lo tanto, el método *k-means* funciona en el espacio vectorial generado, pero no necesariamente funcionaría sin el mapeo del método espectral. Por otra parte, el método espectral también permite determinar el número de clústeres. Aunque esto último no fue práctico para el tipo de datos analizados y, por lo tanto, fue necesario recurrir a otra metodología para la determinación del número de clústeres, lo cual se detallará más adelante.

2.3.3. Propagación de afinidades

Otra metodología que emplea las matrices de similitud es la conocida como propagación de afinidades. Este método se basa en la transmisión de “mensajes” entre

pares de datos, los cuales indican la pertinencia de que estos estén correlacionados, de acuerdo con una cierta medida de similitud, s . Los “mensajes” intercambiados se dividen en dos categorías: la responsabilidad, $r(i, k)$ (ec. 48), y la disponibilidad, $a(i, k)$ (ec. 49). La primera es una medida de la evidencia acumulada de que la muestra k debe ser el ejemplar² de i . La segunda es una medida de la evidencia acumulada de que la muestra i debe elegir a k como su ejemplar.

$$r(i, k) \leftarrow s(i, k) - \max\{a(i, k') + s(i, k') \mid k' \neq k\} \quad (48)$$

$$a(i, k) \leftarrow \min\{0, r(k, k) + \sum_{i' \text{ t.q. } i' \neq i, k} r(i', k)\} \quad (49)$$

Por otra parte, las auto-disponibilidades se modifican de acuerdo con la ecuación 50, la cual establece la evidencia acumulada de que el punto k es un ejemplar, de acuerdo con las responsabilidades positivas que le son “asociadas” por otros puntos.

$$a(k, k) \leftarrow \sum_{i' \text{ t.q. } i' \neq k} \max\{0, r(i', k)\} \quad (50)$$

En el primer paso del algoritmo las disponibilidades son cero: $a(i, k) = 0$. Posteriormente, conforme el algoritmo avanza, las disponibilidades de los puntos elegidos como ejemplares serán negativas para aquellos datos que no les son asociados, lo cual decrece los valores efectivos de sus correspondientes similitudes en el cálculo de las responsabilidades (ec. 49). Esto implica que aquellos puntos elegidos como ejemplares serán parcialmente o completamente descartados en el cálculo de las responsabilidades de aquellos puntos para los que no sean ejemplares. Así, eventualmente se termina exacerbando la relación de cada dato con su ejemplar. Finalmente, la evidencia acumulada de los mensajes se condensa en una matriz denominada matriz de criterio:

$$c(i, k) \leftarrow r(i, k) + a(i, k) \quad (51)$$

En esta última matriz pueden identificarse fácilmente los datos ejemplares y sus datos asociados. Tanto las columnas como las filas de la matriz de criterio conservan las etiquetas de cada dato. Así, las columnas asociadas a los valores más grandes en cada fila de la matriz de criterio corresponden a los datos ejemplares y aquellas filas con los mismos datos ejemplares se asocian a un mismo clúster. Así, de manera

²Un ejemplar es un dato representativo de un cierto subconjunto de datos, tal que en términos de las distancias este es un punto central en un clúster.

iterativa se encuentran los clústeres conformados por datos ejemplares y aquellos que les son asociados (Frey y Dueck 2007, Thavikulwat 2008).

Es importante remarcar que el método de propagación de afinidades tampoco depende de especificar el número de clústeres (Xu y Tian 2015, Frey y Dueck 2007, Thavikulwat 2008). Sin embargo, se puede dar preferencia a ciertos datos para ser elegidos como ejemplares, lo cual afecta el número de clústeres formados de manera indirecta. Esto se logra al modificar los valores de la diagonal de la matriz de similitud ($s(k, k)$), los cuales se denominan valores de *preferencia*, ya que datos con valores más grandes de auto-similitud serán probablemente elegidos como ejemplares (ec. 48). Si a todos los datos se les asocia un valor común de auto-similitud, valores cercanos a la media de las similitudes resultarán en un número moderado de clústeres, mientras que valores pequeños resultarán en pocos clústeres (Frey y Dueck 2007). Así, se tiene una dependencia indirecta entre el número de agrupaciones y los propios hiperparámetros del método, lo cual puede ser complicado de controlar.

2.3.4. Ordenamiento de puntos para identificar la estructura de clústeres

Otra metodología de agrupamiento que puede ser adecuada para el tipo de datos analizados en este trabajo es la conocida como ordenamiento de puntos para identificar la estructura de clústeres (*OPTICS*, por sus siglas en inglés)(Ankerst et al. 1999). Esta metodología utiliza como información de entrada matrices de distancia, a diferencia de los dos métodos anteriores que emplean matrices de similitud. El algoritmo *OPTICS* está basado en otro método conocido como agrupación espacial basada en densidad de aplicaciones con ruido (*DBSCAN*, por sus siglas en inglés) (Ester et al. 1996, Schubert et al. 2017). En general, como el nombre anterior lo indica, este tipo de algoritmos son conocidos como métodos de densidad. Estos consideran que los clústeres estarán definidos por regiones con densidades particulares, las cuales difieren considerablemente de aquellas en regiones fuera de los clústeres. De acuerdo con Ester et al. (1996), *DBSCAN* consiste en encontrar para cada punto el número de vecinos cercanos dentro de un cierto radio ϵ , tal que la vecindad de un punto p está dada por la siguiente expresión:

$$N_\epsilon(p) = \{q \in D | d(p, q) \leq \epsilon\} \quad (52)$$

donde d es cualquier distancia y q cualquier otro punto en el conjunto de datos. Si el número de puntos cercanos es mayor a un hiperparámetro *MinPts*, entonces el punto es considerado un punto central o *core point*:

$$Card(N_\epsilon(p)) \geq MinPts \quad (53)$$

El mismo procedimiento se repite para todos los puntos cercanos al punto central previamente analizado. Si existe un punto que sea cercano a otro punto central, pero no es él mismo un punto central, entonces este es identificado como un punto de frontera, el cual es asociado al mismo subconjunto de datos, pero no es considerado en la búsqueda de puntos. Esto es, el algoritmo itera solo sobre los puntos centrales cercanos. Una vez que un caso ha sido analizado y no se encuentran más puntos centrales cercanos, entonces el algoritmo busca los puntos en el conjunto completo de datos que no hayan sido considerados todavía. Si para un cierto dato no se encuentran puntos cercanos ($d(p, q) > \epsilon \forall q$), este simplemente no es asociado a ningún clúster. De esta manera se genera al final un conjunto de clústeres (conformados por puntos centrales y de frontera) y de puntos considerados como ruido (datos que no pertenecen a ningún clúster).

Una de las principales desventajas del algoritmo *DBSCAN* es que este considera un único radio ϵ , lo cual equivale a considerar clústeres de densidades similares. Esto último no necesariamente se cumple. Por lo tanto, se propuso el método *OPTICS*, el cual considera de manera simultánea diferentes densidades al variar la distancia ϵ (Ankerst et al. 1999). Esta metodología incorpora dos conceptos: *alcanzabilidad* y *distancia-central*. La distancia-central, *core-d*, de un punto p depende de que este sea un punto central. De otra manera, esta distancia queda indefinida, tal que

$$core - d_{\epsilon, MinPts}(p) = \begin{cases} Indefinida, & \text{si } Card(N_\epsilon(p)) < MinPts \\ MinPts - d(p), & \text{si } Card(N_\epsilon(p)) \geq MinPts \end{cases} \quad (54)$$

donde $MinPts - d(p)$ es la distancia más corta entre el punto p y algún otro en su vecindario, tal que el punto p seguiría siendo central si $\epsilon = MinPts - d(p)$. Por otra parte, la *alcanzabilidad* de un punto p queda definida respecto a un punto central o . Así, si la distancia entre estos puntos es mayor a la distancia-central de p , la alcanzabilidad toma este valor. De lo contrario, la *alcanzabilidad* toma el valor de la *distancia-central* de o :

$$\begin{aligned}
\text{alcanzabilidad} - d_{\epsilon, \text{MinPts}}(p, o) = \\
\begin{cases} \text{Indefinida,} & \text{si } \text{Card}(N_{\epsilon}(o)) < \text{MinPts} \\ \max(\text{core} - d(o), d(o, p)), & \text{si } \text{Card}(N_{\epsilon}(o)) \geq \text{MinPts} \end{cases} \quad (55)
\end{aligned}$$

Así, la *alcanzabilidad* representa la distancia ϵ mínima tal que p se encontraría en el vecindario de o , considerando que este último es un punto central (Ankerst et al. 1999).

De manera general, el algoritmo *OPTICS* itera de manera similar al método *DBSCAN*. Esto es, se elige algún punto de manera aleatoria, se calcula su distancia central y la *alcanzabilidad* de sus vecinos cercanos, dados los hiperparámetros ϵ y *Minpts*. Posteriormente, se itera sobre los puntos centrales más cercanos hasta que no se encuentren más de estos en la vecindad de los puntos analizados. En particular para el método *OPTICS*, los valores de *alcanzabilidad* se van actualizando. Al calcular las distancias entre un nuevo punto central y sus vecinos cercanos, si alguno de los puntos tiene una menor distancia respecto al nuevo punto central que respecto al previo, entonces su valor de *alcanzabilidad* se actualiza por la distancia más corta. Además, los puntos se van ordenando de menor a mayor *alcanzabilidad* conforme se itera sobre los puntos centrales cercanos. Así, al final se obtiene una lista de las *alcanzabilidades* mínimas de todos los datos y su ordenamiento, tal que puntos cercanos en la lista final son puntos cercanos en el espacio de los datos. De esta manera es posible identificar clústeres, ya que los valores de *alcanzabilidad* deben ser cercanos para datos de un mismo clúster. Asimismo, datos de diferentes clústeres tendrán una gran diferencia entre sus valores de *alcanzabilidad*. Esto puede analizarse al graficar los valores de *alcanzabilidad* de todos los puntos, manteniendo el ordenamiento final del algoritmo (Ankerst et al. 1999). También, es posible definir un hiperparámetro ξ tal que, entre valores consecutivos, si la diferencia en *alcanzabilidad* se encuentra por debajo de este hiperparámetro, entonces los datos se asocian a un mismo clúster. Esto último es lo implementado en la librería *scikit-learn* (Pedregosa et al. 2011).

2.3.5. Validación de las agrupaciones

De acuerdo con lo anterior, existen diferentes métodos de agrupamiento que se pueden emplear. Sin embargo, no todos necesariamente generan los mejores resultados. Así, existen diferentes métricas para validar las agrupaciones resultantes de este tipo de metodologías. Muchas de estas dependen de conocer de antemano algunas

agrupaciones y por comparación con los resultados se validan los métodos (Brun et al. 2007). Sin embargo, en el caso de ignorar esta información, lo cual es lo usual en métodos de agrupamiento, se suele utilizar el coeficiente o puntaje de silueta (*silhouette score*, en inglés) (Rousseeuw 1987, Starczewski y Krzyżak 2015). Esta medida consiste en comparar las distancias de los datos dentro de sus agrupaciones con respecto a sus distancias a un clúster próximo. Para cada dato este parámetro se define de la siguiente manera:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (56)$$

donde a es la distancia promedio entre la muestra i y los demás datos dentro del mismo clúster, b es la distancia promedio entre la muestra y los datos del clúster más cercano. Esta medida toma valores entre -1 y 1. Valores cercanos a -1 implican que el dato pertenece a otro clúster, valores cercanos a 0 indican que existe otro clúster al que también podría pertenecer y valores cercanos a 1 revelan una buena agrupación (Rousseeuw 1987). Así, para validar los clústeres se suele reportar el promedio sobre todos los datos de este parámetro. Además, se ha mostrado que el promedio del coeficiente de silueta permite identificar correctamente el número óptimo de clústeres para diferentes métodos de agrupamiento y sin importar la forma que adopten las agrupaciones (Starczewski y Krzyżak 2015).

Sin embargo, cabe mencionar que un coeficiente de silueta promedio muy cercano a 1 no implica necesariamente una agrupación óptima. El promedio de este parámetro puede estar sobrepesado por algunos valores iguales o muy cercanos a 1. Esto último puede ocurrir si se obtienen agrupaciones con uno o muy pocos datos. Por lo tanto, un valor alto del puntaje de silueta promedio podría indicar una mala agrupación, ya que se tendrían datos que no forman realmente parte de un clúster o agrupaciones con poblaciones que no son representativas. Así, es necesario confirmar el resultado al analizar el número de datos en cada agrupación y, así, verificar que no se cumpla alguno de los casos extremos previamente mencionados.

Un análisis alternativo al promedio del coeficiente de silueta para la validación de los clústeres es graficar el valor de la medida para cada dato y de manera separada para cada clúster (Rousseeuw 1987). Este tipo de representaciones permite conocer más detalles sobre las agrupaciones, tales como la existencia de datos aislados (con puntaje de silueta negativo), qué tan similares son los datos en cada clúster y qué tan diferentes son las poblaciones de los grupos. Así, esto permite complementar el análisis. En la sección de resultados se mostrarán más detalles y ejemplos de esta representación.

2.4. Métricas en el espacio configuracional

De acuerdo con lo anterior, la aplicación de métodos de agrupamiento depende de una correcta descripción y comparación de los datos u objetos analizados. Asimismo, otros métodos de aprendizaje automático requieren de métricas que permitan distinguir entre datos similares de aquellos que no lo son o para el aprendizaje de campos de fuerza a partir de las representaciones de los datos. En general en química, esto implica tener medidas de moléculas o materiales que permitan distinguir entre ambientes de empaquetamiento, ambiente químico, topología, actividad biológica, reactividad, etc. (Onat et al. 2020, Langer et al. 2022). En particular, las medidas de disimilitud (métricas) o de similitud entre redes cristalinas moleculares en el espacio configuracional, también referidas como representaciones de densidad de la vecindad atómica, son relevantes en la actualidad. El espacio configuracional es un espacio matemático relacionado con todos los posibles arreglos (configuración) de moléculas o átomos en estructuras cristalinas, tal que métricas en este espacio miden la disimilitud entre configuraciones de los compuestos químicos en las redes cristalinas (Sadeghi et al. 2013). Una gran cantidad de medidas se han propuesto, aunque cabe mencionar que no todas se han enfocado en la descripción puramente geométrica de estos sistemas (Onat et al. 2020, Langer et al. 2022, Sadeghi et al. 2013). Entre las medidas más usuales se encuentran el solapamiento suave de posiciones atómicas (SOAP, por sus siglas en inglés) (Bartók et al. 2013), el cual describe los ambientes atómicos en términos de la expansión de funciones gaussianas centradas en los átomos; la representación tensorial de muchos cuerpos (MBTR, en inglés), la cual codifica diferentes descriptores geométricos en un tensor (Huo y Rupp 2022); la matriz de Coulomb (Rupp et al. 2012), la cual representa sistemas a través de los inversos de las distancias entre pares de átomos; entre otras. Así, existe una gran variedad de descriptores espaciales aplicables a sistemas cristalinos; sin embargo, de acuerdo con lo presentado en la introducción, se profundizará solo en la descripción de las configuraciones de cristales moleculares. En consecuencia, se abordarán solo dos métricas de las reportadas en la literatura: la raíz de la desviación cuadrática media (*RMSD*) y el descriptor de coordenadas relativo (*RCD*), ya que estas han sido relevantes en análisis similares (Sadeghi et al. 2013, Li et al. 2018) y, además, se pudieron emplear directamente con la metodología propuesta en este trabajo.

Una manera de comparar las configuraciones de dos sistemas consiste en superponerlos y medir las diferencias entre sus coordenadas. Así, resulta necesario conocer la transformación que logre el traslape de los sistemas de manera óptima. Esto también se puede entender como la transformación de un conjunto de coordenadas de un marco de referencia a otro. Si se conocen dos conjuntos de puntos en dos es-

pacios cartesianos diferentes, entonces se puede buscar la transformación que lleve los puntos del primer conjunto al segundo de manera precisa; es decir, que genere el mejor traslape entre los puntos equivalentes al aplicar la transformación. Esto se conoce como el problema de la transformación de Helmert (Chang 2015, Zeng et al. 2018). Este tipo de problema se resuelve de manera general al minimizar una función de error respecto a los parámetros asociados a las traslaciones y rotaciones de los conjuntos de coordenadas, lo cual se ha tratado empleando tanto representaciones matriciales, como cuaterniones y cuaterniones duales (Coutsias et al. 2004, Sadeghi et al. 2013, Horn et al. 1988, Flower 1999, Ioannidou y Pantazis 2020, Zeng et al. 2018). La idea general, reside en minimizar la raíz de la desviación cuadrática media (RMSD, por sus siglas en inglés). Así, se busca que las distancias entre los puntos obtenidos al aplicar la transformación y aquellos en la referencia a la que se transforma sean mínimas. Este tipo de métodos requiere de algoritmos iterativos o resolución de eigen-ecuaciones para determinar cada transformación óptima. Sin embargo, si para cada sistema entre los que se busca la transformación óptima se asocian solo cuatro puntos representativos, el problema de Helmert tiene solución analítica. Esto es, si se tienen cuerpos rígidos, por ejemplo moléculas con los mismos átomos y cuyos enlaces puedan considerarse iguales, entonces puede representarse a cada sistema o molécula por su centro geométrico y una triada de vectores ortonormales. En consecuencia, la transformación de Helmert tiene solución analítica y puede obtenerse empleando cuaterniones o cuaterniones duales (Horn 1987), lo cual se detallará en las siguientes secciones. Evidentemente, esta solución no será igual al método de minimización del RMSD y, por lo tanto, puede llevar a errores relativamente grandes en algunos puntos o átomos; sin embargo, como se mostrará más adelante, para moléculas pequeñas (pocos átomos) el error obtenido con la aproximación del cuerpo rígido es mínimo.

2.4.1. Cuaterniones duales y su aplicación en el desplazamiento de cuerpos rígidos

Los cuaterniones duales son una herramienta matemática que permite rotar y trasladar objetos en el espacio en pocos pasos y de manera computacionalmente óptima. En particular, esta herramienta permite realizar este tipo de transformaciones de Helmert con una gran precisión, comparando con respecto a usar ángulos de Euler (Ioannidou y Pantazis 2020). Esta herramienta ha sido recientemente implementada en química y ha sido muy útil en modelado molecular, puesto que este ente matemático facilita el estudio del espacio de configuraciones (Karney 2007). Además, el uso de los cuaterniones duales ha permitido mejorar la determinación de la estructura secundaria de proteínas (Hanson et al. 2011, Adasme-Carreño et al.

2021). Así, debido a que en este proyecto se requiere estudiar las posiciones relativas de moléculas en un cristal (*cf.* sección 1), el uso de cuaterniones duales puede facilitar este estudio. Por lo tanto, a continuación se presentan algunas propiedades relevantes de los cuaterniones duales.

Previo a definir un cuaternión dual se debe precisar, evidentemente, qué son los cuaterniones. Un cuaternión q se define como la suma de un escalar q_0 y un vector $\vec{q} = (q_1, q_2, q_3)$ (Jia s.f.), sea

$$q = q_0 + \vec{q} = (q_0, q_1, q_2, q_3) \quad (57)$$

Así, la suma de cuaterniones se realiza por componentes (ec. 58), como en el caso de los vectores, la cual es asociativa y conmutativa. En el caso del producto, este se define de acuerdo con la ecuación 59, el cual no es conmutativo.

$$p + q = (p_0 + q_0) + (\vec{p} + \vec{q}) = (p_0 + q_0, p_1 + q_1, p_2 + q_2, p_3 + q_3) \quad (58)$$

$$pq = p_0q_0 - \vec{p} \cdot \vec{q} + p_0\vec{q} + q_0\vec{p} + \vec{p} \times \vec{q} \quad (59)$$

El conjugado de un cuaternión se define como

$$q^* = q_0 - \vec{q} \quad (60)$$

Tal que,

$$(pq)^* = q^*p^* \quad (61)$$

Por otra parte, una manera alternativa de definir un cuaternión es

$$q^R = \cos \frac{\Theta}{2} + \hat{u} \sin \frac{\Theta}{2} \quad (62)$$

donde $\Theta \in [0, \pi]$ y $\hat{u}$ es un vector unitario. Cabe mencionar que este cuaternión, q^R , tiene la peculiaridad de ser unitario; es decir, $|q^R| = \sqrt{q^R q^{R*}} = 1$. Un cuaternión expresado de esta manera (ec. 62) permite rotar un vector $\vec{v}$ (expresado como cuaternión puro, $v = (0, \vec{v})$), de acuerdo con la ecuación 63 (Jia s.f.). En otras palabras, se obtiene un vector $\vec{v}'$ que resulta de rotar $\vec{v}$ en un ángulo Θ alrededor del vector $\hat{u}$. Así, en un solo ente matemático, un cuaternión, se almacena tanto la información del vector respecto al cuál se está rotando como el ángulo de rotación.

$$v' = qvq^* \quad (63)$$

Por otra parte, los cuaterniones pueden extenderse a cuaterniones duales. Esto parte de la definición de números duales, $\bar{d}$, hecha por Clifford (1871). Estos números son similares a los números complejos, pero en lugar de tener una componente imaginaria se tiene una parte dual:

$$\bar{d} = a + \epsilon b \quad (64)$$

donde a y b son elementos de algún campo algebraico y ϵ es una unidad dual, tal que $\epsilon^2 = 0$. Por lo tanto, el álgebra de estos números es muy similar a la de los números complejos salvo que debe tenerse en cuenta que el producto de las partes duales es nulo (ec. 65).

$$\begin{aligned} \bar{c} \otimes \bar{d} &= (a_1 + \epsilon b_1)(a_2 + \epsilon b_2) \\ &= a_1 a_2 + \epsilon(a_1 b_2 + a_2 b_1) + \epsilon^2 b_1 b_2 \\ &= a_1 a_2 + \epsilon(a_1 b_2 + a_2 b_1) \end{aligned} \quad (65)$$

La definición de un número dual puede extenderse a un vector dual ($\bar{v} = (\bar{v}_1, \bar{v}_2, \bar{v}_3)$), tal que todos sus componentes son números duales. De manera análoga, se puede definir un cuaternión dual,

$$\sigma = p + \epsilon q \quad (66)$$

cuyas dos componentes, p y q , son cuaterniones. El álgebra particular de los cuaterniones duales permite almacenar en ellos la información tanto de una rotación como de una traslación, cabe remarcar que esta última operación no estaba contenida en los cuaterniones. Con el fin de demostrar esto último se detallará sobre algunas de las propiedades relevantes de los cuaterniones duales.

Por una parte, el producto de dos cuaterniones duales sigue la misma álgebra mostrada para el caso de los números duales, tal que

$$\sigma_1 \otimes \sigma_2 = p_1 p_2 + \epsilon(p_1 q_2 + q_1 p_2) \quad (67)$$

Donde los productos de cuaterniones se realizan de acuerdo con la ecuación 59. Por otra parte, dado que tanto los cuaterniones como los números duales tienen

conjugados se pueden definir tres conjugados distintos para los cuaterniones duales:

$$\sigma^\bullet = p - \epsilon q \quad \sigma^* = p^* + \epsilon q^* \quad \sigma^\diamond = p^* - \epsilon q^* \quad (68)$$

Así, el conjugado $\sigma^\diamond$ es una combinación de los otros dos. Este conjugado junto con la definición del producto de cuaterniones duales son las principales herramientas para realizar un desplazamiento rígido (rotación más traslación) de cualquier vector. Con este fin, la rotación de un vector se puede almacenar en un cuaternión unitario, de acuerdo con la ecuación 62. Este último se puede incluir a su vez en un cuaternión dual, al cual también se le añade la traslación con un vector expresado como cuaternión puro, $T = (0, \vec{T})$:

$$\sigma = q^R + \frac{\epsilon}{2} T q^R \quad (69)$$

Así, tomando en cuenta las propiedades previamente descritas, se puede realizar un desplazamiento rígido a partir del cuaternión dual previo (ec.69). Si consideramos que el vector a desplazar es $\vec{v}$, el cual se puede escribir como un cuaternión dual: $v = 1 + \epsilon(0, \vec{v})$; entonces, la rotación y traslación de este vector se pueden compactar en una secuencia de dos productos:

$$\begin{aligned} \sigma \otimes (1 + \epsilon v) \otimes \sigma^\diamond &= (q^R + \frac{\epsilon}{2} T q^R) \otimes (1 + \epsilon v) \otimes (q^{R*} - \frac{\epsilon}{2} (T q^R)^*) \\ &= (q^R + \frac{\epsilon}{2} T q^R) \otimes [q^{R*} + \epsilon(v q^{R*} - \frac{1}{2} q^{R*} T^*)] \\ &= q^R q^{R*} + \epsilon(q^R v q^{R*} - \frac{1}{2} q^R q^{R*} T^* + \frac{1}{2} T q^R q^{R*}) \\ &= 1 + \epsilon[q^R v q^{R*} + \frac{1}{2} (T - T^*)] \\ &= 1 + \epsilon[q^R v q^{R*} + \frac{1}{2} ((0, \vec{T}) - (0, -\vec{T}))] \\ &= 1 + \epsilon[q^R v q^{R*} + (0, \vec{T})] \end{aligned} \quad (70)$$

Así, al realizar esta operación (ec. 70) se obtiene un cuaternión dual que contiene al vector rotado y trasladado. Esto se puede observar directamente en el último renglón de la ecuación anterior puesto que $q^R v q^{R*}$ es la operación de rotación del vector v (expresado como cuaternión), como se describió previamente (ec. 63), al cual se le añade el vector $\vec{T}$ para desplazarlo. Entonces, se obtiene lo mismo que se obtendría al aplicar una matriz de rotación y después añadir el vector $\vec{T}$; sin embargo, en este caso la información está condensada en el cuaternión dual.

Por otra parte, considerando el proceso inverso, si se tiene un cuaternión dual

$\sigma = p + \epsilon q$, se puede recuperar toda la información relevante de un desplazamiento rígido a través del cuaternión de rotación (q^R) y del vector de traslación $\vec{T}$ (Jia s.f.):

$$q^R = p \quad \vec{T} = 2qp^* \quad (71)$$

En conclusión, los cuaterniones duales representan una manera eficiente y compacta de realizar un desplazamiento rígido de cualquier vector así como de almacenar la información asociada a esta operación. Por lo tanto, se considera que esta herramienta matemática puede ser útil para identificar las posiciones relativas (vectores y ángulos) entre moléculas en redes cristalinas, lo cual a su vez puede emplearse para comparar sus configuraciones.

Cuaterniones duales como descriptores espaciales de cuerpos rígidos

De acuerdo con lo discutido en la sección anterior, los cuaterniones duales son una herramienta matemática que permite almacenar rotaciones y traslaciones de manera compacta (Jia s.f.). Por lo tanto, estos pueden ser útiles para describir las posiciones y orientaciones relativas de moléculas en redes cristalinas. Así, en esta sección se detallará sobre el análisis espacial necesario para la descripción configuracional de cristales moleculares implementando cuaterniones duales.

En el análisis teórico-computacional de cristales moleculares es común considerar moléculas rígidas; es decir, se considera que las posiciones relativas entre los átomos de una misma molécula no se modifican por las interacciones en el cristal. Esto difiere de la realidad; sin embargo, es una aproximación aceptable para sistemas en los cuales las interacciones entre las moléculas son débiles (Price 2008), como el caso de la formamida (Reckien y Bredow 2011). Así, bajo esta perspectiva puede analizarse a cada molécula en un cristal como un cuerpo rígido descrito por un conjunto de puntos. Además, si se consideran sistemas de un solo componente, entonces, las moléculas en la red cristalina solo difieren en sus posiciones y orientaciones. Por lo tanto, el análisis espacial de este tipo de sistemas se reduce a considerar las rotaciones y traslaciones necesarias para generar el conjunto completo de moléculas a partir de una de ellas (fig. 1). Esto implica que al aplicar una matriz de rotación, $\mathbf{R}$, sobre un punto, z , del primer conjunto, o molécula, y sumar el vector entre los centros de las coordenadas, o moléculas, $\vec{T}$, se obtiene el punto equivalente en el segundo conjunto, z' (ec. 72).

$$z' = \mathbf{R}z + \vec{T} \quad (72)$$

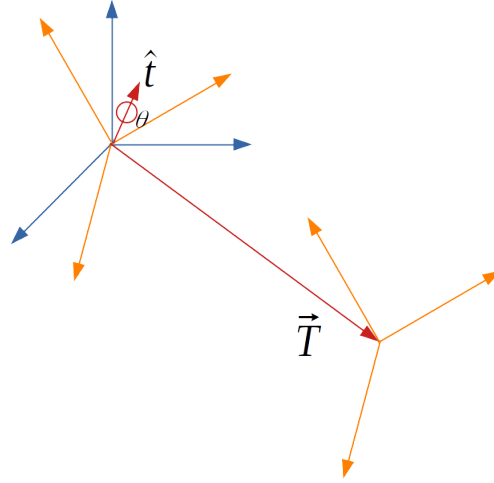


Figura 1: Representación esquemática de la rotación y traslación de un cuerpo rígido, donde $\hat{t}$ es el vector de rotación, θ el ángulo de rotación y $\vec{T}$ el vector entre los centros de coordenadas. El cuerpo rígido es representado por las triadas de vectores en naranja y azul.

Así, el análisis espacial entre dos cuerpos rígidos consiste en determinar la posición entre los centros de sus coordenadas, $\vec{T}$, y la orientación relativa a través de un ángulo, θ , y un vector respecto al cual se aplica la rotación, $\hat{t}$ (fig. 1) (Bottema y Roth 1979, Cap. 1). Estos parámetros se pueden obtener a partir de definir una triada de vectores ortonormales para cada cuerpo rígido. Para esto es necesario seleccionar tres puntos equivalentes en cada conjunto de coordenadas. A partir de esto, se selecciona uno de los puntos como el central y se generan vectores entre cada uno de los restantes y este. Con el propósito de ejemplificar lo anterior, sean $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$, los vectores asociados a tres puntos de un mismo conjunto. Así, considerando que se selecciona al primer vector como el central, se establece lo siguiente:

$$\vec{w}_1 = \vec{v}_2 - \vec{v}_1 \quad \vec{w}_2 = \vec{v}_3 - \vec{v}_1 \quad (73)$$

Con los vectores anteriores se puede generar un conjunto de vectores ortonormales, $\vec{u}$, o base. Esto es, se aplica el método de ortonormalización de Gram-Schmidt (Nachtergaele y Schilling 2021) para generar dos vectores ortonormales y el tercero se obtiene del producto vectorial entre estos:

$$\begin{aligned} \vec{u}_1 &= \vec{w}_1 \\ \vec{u}_2 &= \vec{w}_2 - \frac{\vec{w}_2 \cdot \vec{u}_1}{|\vec{u}_1|^2} \vec{u}_1 & \vec{u}_3 &= \vec{u}_1 \times \vec{u}_2 \end{aligned} \quad (74)$$

donde los vectores u se pueden normalizar simplemente dividiéndolos entre sus magnitudes. Así, una vez que se tiene una base para cada conjunto de coordenadas, o molécula, de interés se puede establecer la rotación relativa entre estas. Considerando que existe un eje de rotación l respecto al cuál se rota la primera base para generar la segunda (Fig. 1), se pueden descomponer los vectores de la primera base ortonormal respecto a esta línea:

$$\vec{u}_i = \vec{u}_i^p + \vec{u}_i^t \quad (75)$$

donde $\vec{u}_i$ es cualquiera de los tres vectores ortonormales, $\vec{u}_i^p$ es su componente perpendicular al eje de rotación y $\vec{u}_i^t$ la componente paralela al mismo. Así, dado que las componentes de las bases ortonormales están relacionadas de acuerdo con la ecuación 72:

$$\vec{u}_i' = \mathbf{R}\vec{u}_i + \vec{T} = \vec{u}_i^{R,p} + \vec{u}_i^{R,t} + \vec{T} \quad (76)$$

donde el superíndice R indica los vectores resultantes de aplicar la matriz de rotación sobre el vector $\vec{u}_i$. Así,

$$\Delta\vec{u}_i = \vec{u}_i' - \vec{u}_i = \vec{u}_i^{R,p} + \vec{u}_i^{R,t} + \vec{T} - \vec{u}_i^p - \vec{u}_i^t \quad (77)$$

Dado que la rotación se realiza en el plano perpendicular al eje de rotación, las componentes paralelas a este no se modifican:

$$\vec{u}_i^{R,t} = \vec{u}_i^t \quad \text{Sea} \quad \Delta\vec{u}_i = \vec{u}_i^{R,p} + \vec{T} - \vec{u}_i^p \quad (78)$$

De manera que al restar estas diferencias para dos vectores ortonormales se tiene lo siguiente:

$$\begin{aligned} \Delta\vec{u}_1 - \Delta\vec{u}_2 &= \vec{u}_1^{R,p} + \vec{T} - \vec{u}_1^p - \vec{u}_2^{R,p} - \vec{T} + \vec{u}_2^p = \vec{u}_1^{R,p} - \vec{u}_1^p - (\vec{u}_2^{R,p} - \vec{u}_2^p) \\ \Delta\vec{u}_1 - \Delta\vec{u}_3 &= \vec{u}_1^{R,p} + \vec{T} - \vec{u}_1^p - \vec{u}_3^{R,p} - \vec{T} + \vec{u}_3^p = \vec{u}_1^{R,p} - \vec{u}_1^p - (\vec{u}_3^{R,p} - \vec{u}_3^p) \end{aligned} \quad (79)$$

Por lo tanto, los vectores anteriores (ecs. 79) se encuentran en el plano perpendicular al eje de rotación, ya que todas sus componentes son ortogonales a este. Así, basta con realizar el producto cruz entre estos vectores para generar el vector de rotación, $\hat{t}$:

$$\vec{t} = (\Delta\vec{u}_1 - \Delta\vec{u}_2) \times (\Delta\vec{u}_1 - \Delta\vec{u}_3) \quad (80)$$

Una vez conocido el vector de rotación, se normaliza y a partir de este se calcula el ángulo de rotación, θ . Dado que se tienen vectores equivalentes en ambos marcos de referencia, se puede obtener el ángulo entre un par de estos al proyectarlos sobre el plano perpendicular a $\hat{t}$:

$$\vec{u}_i^p = \vec{u}_i - (\vec{u}_i \cdot \hat{t})\hat{t} \quad (81)$$

Así,

$$\theta = \arccos\left(\frac{\vec{u}_i^p \cdot \vec{u}_j^p}{|\vec{u}_i^p||\vec{u}_j^p|}\right) \quad (82)$$

De esta manera se puede obtener el ángulo, θ , y el vector, $\hat{t}$ de rotación entre un par de moléculas. Esto a su vez puede condensarse en un cuaternión, q^R , de acuerdo con la ecuación 62. Tal que, este tipo de cuaternión cumple con la ecuación 83 para cualquiera de las coordenadas equivalentes del par de cuerpos rígidos, o moléculas, (Jia s.f.).

$$z' = qzq^* \quad (83)$$

Por otra parte, hasta el momento solo se ha considerado la rotación y es necesario recurrir a los cuaterniones duales para incluir la parte traslacional, ya que estos permiten incluir la posición relativa entre dos moléculas, $\vec{T}$ (fig. 1), de acuerdo con la ecuación 69. Así, de acuerdo con las definiciones anteriores, la rotación y traslación de un vector $\vec{v}$ están dadas por la siguiente expresión:

$$1 + \epsilon(0, \vec{v}') = \sigma(1 + \epsilon(0, \vec{v}))\sigma^\diamond \quad (84)$$

tal que $\vec{v}$ es un punto en el primer conjunto de coordenadas y $\vec{v}'$ es el punto correspondiente en el segundo conjunto. En términos de la comparación de dos moléculas, estos vectores pueden ser coordenadas de átomos equivalentes en cada molécula. Así, σ definido como en la ecuación 69 permite almacenar toda la información necesaria para transformar un conjunto de coordenadas en otro. En otras palabras, en este tipo de cuaternión dual se condensa la información espacial relevante entre un par de moléculas.

Por otra parte, es importante señalar que las coordenadas que se obtengan al

realizar la transformación con un cuaternión dual (ec. 84) estarán referidas al centro de coordenadas del cual se parta (Jia s.f.). Por lo tanto, al momento de hacer una comprobación es necesario sumar a las coordenadas resultantes la posición de este último. De esta manera se obtienen las coordenadas que deben coincidir con las correspondientes al segundo marco de referencia a partir del cual se generó el cuaternión dual de acuerdo con lo descrito previamente. Es decir, es necesario asegurar que todas las coordenadas se encuentren bajo el mismo marco de referencia.

Comparación de cuaterniones duales

De acuerdo con lo descrito en las secciones anteriores, es posible calcular para una cierta estructura cristalina los cuaterniones duales entre una de sus moléculas, tomada como referencia, y todas aquellas que la rodeen. Sin embargo, esta información es basta y difícil de analizar sobre todo si consideramos que los sistemas tienen una gran cantidad de moléculas. Por lo tanto, con el fin de equiparar los diferentes ambientes químicos que puedan presentarse en los cristales moleculares es necesario comparar los conjuntos de cuaterniones duales que los describen. Sin embargo, no existe un método estándar en la literatura para la comparación de cuaterniones duales, hasta donde sabemos. Por lo tanto, es necesario proponer parámetros de comparación basados en el álgebra de estos elementos. Esto se detallará en la metodología; sin embargo, resulta importante recordar algunas propiedades de los cuaterniones y cuaterniones duales, las cuales serán utilizadas más adelante:

- El segundo conjugado de un cuaternión dual definido como en la ecuación 69 se expresa como $\sigma^* = q_r^* + \epsilon \frac{1}{2}(Tq_r)^*$ y representa la transformación inversa del cuaternión sin conjugar (Jia s.f.).
- El producto de un cuaternión dual unitario, como los definidos en la sección anterior, y su segundo conjugado (σ^*) cumple con la ecuación 85 (Jia s.f.).

$$\sigma \otimes \sigma^* = 1 \quad (85)$$

- El producto de un cuaternion, q , y su conjugado, q^* , es igual a su magnitud al cuadrado (ec. 86) (Jia s.f.)

$$qq^* = q^*q = |q|^2 \quad (86)$$

2.4.2. Raíz del error cuadrático medio (RMSD)

Una manera alternativa de comparar moléculas es simplemente medir la distancia euclidiana entre todos los pares de átomos, considerando que las coordenadas están referidas a la misma base. Así, la comparación de un par de moléculas puede condensarse en una medida de la raíz cuadrada de la desviación cuadrática media (RMSD, ec. 87) entre las coordenadas de ambas estructuras. Así, esto permite tener una medida bien establecida, la distancia euclidiana. La idea general es considerar que a través de una rotación y una traslación se sobreponen los sistemas a comparar, tal que las distancias entre los átomos equivalentes serán nulas si ambas moléculas son iguales. Este tipo de medida ha sido ampliamente usada en química, tanto para la comparación de sólidos como de proteínas (Kufareva y Abagyan 2012, Mechelke y Habeck 2010, Hönig et al. 2022, Sadeghi et al. 2013, Bell y Zhang 2019). Entonces, de acuerdo con lo mencionado previamente, la determinación del *RMSD* depende de conocer la transformación óptima que traslapa los sistemas comparados, lo cual corresponde a minimizar la siguiente función:

$$RMSD(\alpha, \beta) = \frac{1}{\sqrt{N}} \min_{\mathbf{R}, \mathbf{P}} \left(\sum_{i=1}^n |\vec{v}_i^\alpha - \mathbf{R} \vec{v}_i^\beta \mathbf{P}|^2 \right)^{1/2} \quad (87)$$

donde $\vec{v}^\alpha$ son las coordenadas de los puntos o átomos en la molécula α , $\vec{v}^\beta$ las coordenadas equivalentes en la molécula β , referidas a sus centros geométricos correspondientes; n es el número de átomos en cada molécula; $\mathbf{R}$ es una matriz de rotación que traslapa los átomos de β con los del α y $\mathbf{P}$ es una matriz que intercambia los índices de los átomos con el fin de considerar todos los posibles acomodos. Sin embargo, en el caso de comparar moléculas que solo difieren en su orientación en el espacio, el *RMSD* puede emplearse como una medida directa de la variación de las coordenadas entre estas:

$$RMSD(\alpha, \beta) = \frac{1}{\sqrt{N}} \min_{\mathbf{R}} \left(\sum_{i=1}^n |\vec{v}_i^\alpha - \mathbf{R} \vec{v}_i^\beta|^2 \right)^{1/2} \quad (88)$$

donde se mide la distancia euclidiana únicamente entre átomos equivalentes.

En el caso de cristales moleculares, pueden compararse las moléculas de dos estructuras cristalinas diferentes por pares. Aunque, en este caso será necesario establecer la transformación que traslapa ambos sistemas (87), lo cual se planteará en la metodología. Esta métrica permite tener un punto de comparación más riguroso y no arbitrario, ya que emplea directamente la distancia euclidiana. Sin embargo, este método resulta mucho más costoso dada la comparación de todos los pares de

átomos para cada par de moléculas. Aun así, captura de manera directa el análisis que se busca lograr con las otras métricas aquí propuestas; es decir, mide la similitud entre las posiciones de los átomos de las moléculas. Por lo tanto, esta medida sirve como un buen punto de comparación, tal que se espera poder capturar información similar con las medidas propuestas con cuaterniones duales.

Además, cabe mencionar que el *RMSD* es en general una medida estadística de dispersión de los datos, equivalente a la desviación estándar. Por lo tanto, la ecuación 87 puede entenderse también como una medida de la dispersión de las distancias respecto a un cierto punto de referencia, lo cual fue otra manera en que se empleó esta medida en este proyecto. Esto último también se plantea en la metodología.

2.4.3. Descriptor de coordenadas relativo

Una medida alternativa para comparar configuraciones de cristales moleculares es el descriptor de coordenadas relativo (RCD, por sus siglas en inglés), propuesto por Li et al. (2018). Esta también depende de describir a cada molécula en los sistemas cristalinos con una triada de vectores ortonormales; es decir, considera la aproximación de la descripción de moléculas como cuerpos rígidos. Así, se describe a cada cristal molecular con un conjunto de pares de vectores asociados a cada molécula en los sistemas:

$$\vec{R} = \{(\vec{T}_1, \vec{Q}_1), \dots, (\vec{T}_M, \vec{Q}_M)\} \quad (89)$$

donde $\vec{T}$ es el vector entre el centro geométrico de alguna de las moléculas y el correspondiente a la de referencia; $\vec{Q}$ son vectores cuyas entradas son los productos puntos entre los vectores ortormales de cada molécula y los equivalentes en la de referencia (fig. 2). Cabe mencionar que la elección de la molécula de referencia no está detallada en el artículo en el cual se propone esta medida Li et al. (2018); por lo tanto, se mantendrá de manera arbitraria a la molécula de referencia como la primera molécula en la lista de coordenadas de cada sistema, por consistencia con la metodología descrita más adelante. La comparación entre los conjuntos de vectores (ec. 89) para cada par de estructuras genera una matriz $\mathbf{R}$, cuyas entradas están dadas por la siguiente expresión:

$$r_{k,l} = \frac{|\vec{T}_k^1 - \vec{T}_l^2|^2}{|\vec{T}_k^1| |\vec{T}_l^2|} - \frac{1}{3} |\vec{Q}_k^1 - \vec{Q}_l^2|^2 \quad (90)$$

Finalmente, las matrices $\mathbf{R}$ generadas para cada par de estructuras son condensadas en un único valor de acuerdo con la siguiente expresión:

$$D_{ij}^{RCD} = \sum_{k=1}^l \min_k(\mathbf{R}_{ij}) \quad (91)$$

donde *min* implica una búsqueda de mínimos sobre toda la matriz $\mathbf{R}_{ij}$ bajo la restricción de que los valores mínimos encontrados no correspondan a la misma columna o al mismo renglón. Lo anterior equivale al algoritmo de búsqueda del mínimo promedio (*bmp*) descrito más adelante en la metodología.

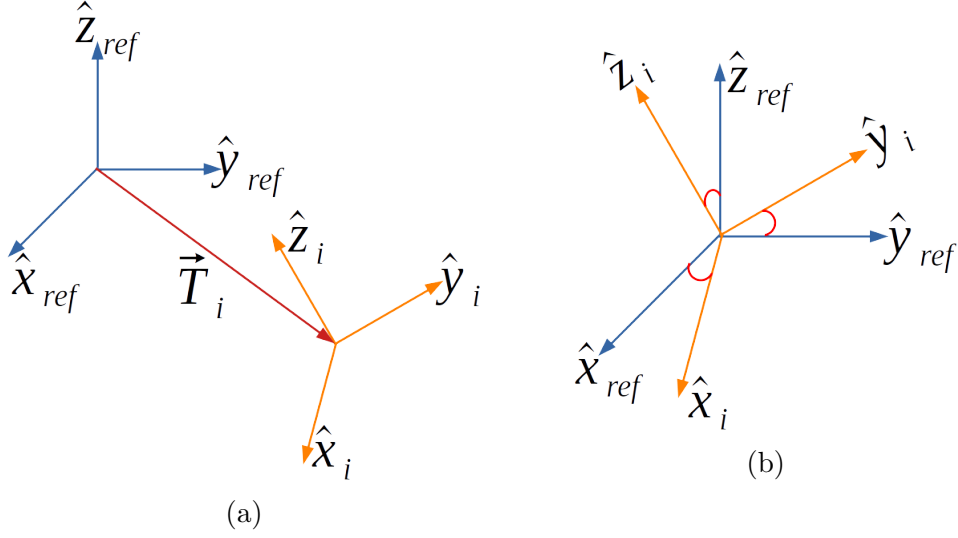


Figura 2: Representación esquemática de la descripción vectorial de moléculas en redes cristalinas propuesta con el descriptor de coordenadas relativo, donde (a) representa el vector de posición, $\vec{T}_i$, y (b) el vector de orientación relativa, $\vec{Q}_i$.

2.5. Cristales moleculares analizados

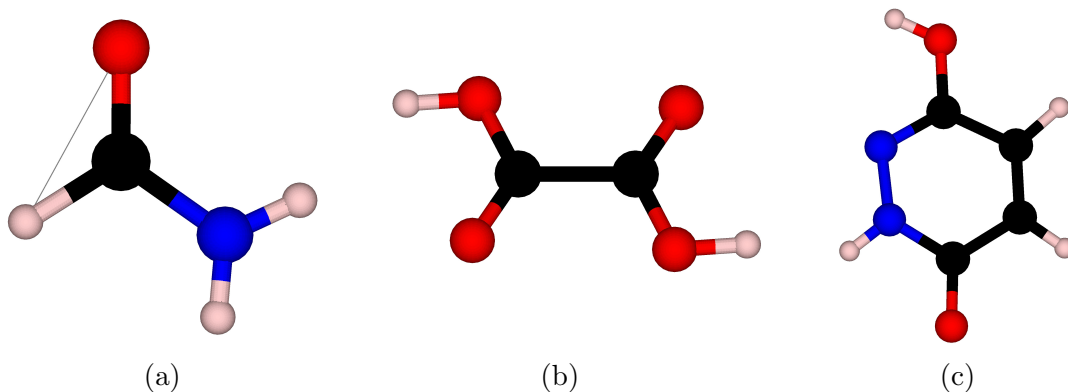


Figura 3: Representaciones 3D de las geometrías moleculares de (a) formamida, (b) ácido oxálico e (c) hidrazida maleica.

Para generar un primer conjunto de estructuras polimórficas se seleccionó la formamida (fig. 3a) como sistema modelo, ya que esta molécula posee una estructura relativamente simple, lo cual permite reducir el costo computacional. Además, la formamida posee una estructura molecular similar a la de un aminoácido y, por lo tanto, es capaz de formar un enlace similar al peptídico. Esto permitirá, de ser concluyente el primer análisis, estudiar *a posteriori* dímeros de esta molécula con el fin de acercarse al estudio de péptidos bajo el método que en este proyecto se busca establecer. Asimismo, la aproximación de cuerpo rígido es aplicable para esta molécula (Reckien y Bredow 2011). Por otra parte, de acuerdo con lo reportado en la literatura, la formamida cristaliza en dos estructuras polimórficas, denominadas fases α y β (Gajda y Katrusiak 2011, Ladell y Post 1954*b,a*). Sin embargo, la fase β existe a presiones muy altas y en este modelo no se abordará la presión como variable. Por lo tanto, se trabajará únicamente considerando la fase α de la formamida, la cual además es la más reportada en la literatura (Ladell y Post 1954*a*, Reckien y Bredow 2011, Itoh y Shimanouchi 1972). Así, se espera que, al existir una única estructura cristalina en condiciones estándar de temperatura y presión, la identificación de la estructura más estable sea relativamente simple. Los datos experimentales de la fase α se presentan en la tabla 1.

Por otra parte, se consideraron las moléculas de la hidrazida maleica y del ácido oxálico. Ambos compuestos exhiben polimorfismo a condiciones estándar de temperatura y presión, con 3 y 2 polimorfos, respectivamente. Estos sistemas han sido ampliamente caracterizados (Reilly y Tkatchenko 2013, Dolgonos et al. 2019, Bhattacharya et al. 2013, Thalladi et al. 2000, Cradwick 1976, Katrusiak 1993, 2001) y empleados en la validación de métodos de predicción de estructura cristalina (McDonagh et al. 2019). Asimismo, estos sistemas fueron seleccionados con el fin de

complementar el análisis de la formamida, ya que estos sí presentan polimorfismo. Los datos cristalográficos empleados se muestran en la tabla 1. Además, cabe mencionar que las estructuras cristalinas del ácido oxálico serán denotadas como *OA* y de la hidrazida maleica como *MH*, diferenciando los polimorfos de acuerdo con las etiquetas mostradas en la tabla 1.

Es importante mencionar que los datos experimentales empleados (tab. 1) no son los más actuales ni los más precisos. Sin embargo, las estructuras cristalinas determinadas experimentalmente se emplearon únicamente para aplicar y validar la metodología de análisis de la configuración de moléculas en redes cristalinas desarrollada en este proyecto. Para esto las estructuras cristalinas experimentales fueron comparadas con estructuras que fueron generadas de manera aleatoria, las cuales no fueron de mínima energía ni fueron corregidas por expansión térmica. En ningún momento se buscó reproducir los datos experimentales. Por lo tanto, la precisión de los datos experimentales empleados en esta tesis fue irrelevante.

Estructura	Formamida- α	$OA - \alpha$	$OA - \beta$	$MH1$	$MH2$	$MH3$
a / Å	3.69(1)	6.4928(1)	5.3266(13)	5.83(2)	6.891(2)	6.6070(15)
b / Å	9.18(3)	6.0600(2)	6.0126(10)	5.78(2)	9.674(2)	6.9070(7)
c / Å	6.87(2)	7.8030(3)	5.4456(13)	7.31(2)	6.946(1)	10.539(3)
α (°)	90.00	90.00	90.00	79.0(3)	90.00	90.00
β (°)	98.00(25)	90.00	115.80(3)	99.5(3)	100.07(2)	104.00(4)
γ (°)	90.00	90.00	90.00	107.2(3)	90.00	90.00
V / Å ³	230.45(3)	307.02(2)	157.01(6)	229.4(13)	455.9(2)	466.68(17)
Z	4	4	2	2	4	4
Sistema cristalino	Monoclínico	Ortorrómbico	Monoclínico	Triclínico	Monoclínico	Monoclínico
Grupo espacial	$P2_1/n$	$Pbca$	$P2_1/c$	$P\bar{1}$	$P2_1/c$	$P2_1/n$
Temperatura / K	223	130	298(2)	295	295	293(2)
Identificador en la CSD	-	OXALAC05	OXALAC07	MALEHY10	MALEHY01	MALEHY12
Referencia	Ladell y Post (1954a)	Thalladi et al. (2000)	Bhattacharya et al. (2013)	Cradwick (1976)	Katrusiak (1993)	Katrusiak (2001)

Tabla 1: Datos cristalográficos de los compuestos analizados y considerados como sistemas de referencia para la implementación y validación de la metodología desarrollada en esta tesis. Los polimorfos del ácido oxálico están etiquetados como OA y los de la hidrazida maleica como MH . Se indican las claves de referencia en la base de datos CSD de los archivos empleados en este trabajo.

2.6. Generación aleatoria de estructuras cristalinas moleculares

En una primera etapa de este proyecto se debe generar un conjunto vasto de posibles configuraciones cristalinas de un cierto compuesto con el fin de identificar posteriormente las estructuras con configuraciones similares con el método a desarrollar. Así, se empleó el programa *Genarris* desarrollado por Li et al. (2018), el cual permite generar un conjunto de posibles estructuras cristalinas de manera aleatoria, dada una cierta estructura molecular. Con el fin de esclarecer el tipo de estructuras analizadas se presenta en esta sección un resumen de esta metodología:

- La información de entrada del programa *Genarris* consiste en la geometría molecular del compuesto a analizar, dada en coordenadas cartesianas x, y, z ; el número de moléculas en las celdas unitarias a generar (Z), el volumen promedio de las celdas ($\bar{V}$), la desviación estándar respecto al volumen promedio (σ_V), el número máximo de estructuras y el número máximo de iteraciones por cada grupo espacial considerado.
- El primer paso del algoritmo consiste en determinar los grupos espaciales compatibles con el número de moléculas indicado para las celdas unitarias. Para esto se buscan los grupos de simetría entre los 230 definidos en cristalografía (Hahn 2002) cuyas posiciones generales de Wyckoff³ tienen una multiplicidad⁴ igual al número de moléculas indicado, de acuerdo con las definiciones estándar de los grupos espaciales del servidor cristalográfico de Bilbao (Aroyo et al. 2006). Así, el algoritmo itera sobre los grupos espaciales adecuados.
- Para un dado grupo espacial, se genera una celda unitaria con parámetros de red elegidos de manera aleatoria, aunque asegurando que el volumen de la celda resultante se encuentre dentro del rango establecido por el volumen promedio y su desviación estándar ($\bar{V} \pm \sigma_V$).
- Una vez generada la celda unitaria, se posiciona en ella la molécula dada en el archivo de entrada. Primero, se coloca el centro geométrico de la molécula en el origen, esta se rota con un vector de rotación elegido de manera aleatoria sobre una esfera unitaria y, finalmente, se mueve el centro geométrico de la

³Las posiciones de Wyckoff representan puntos en la celda que son invariantes ante la aplicación de ciertos elementos de simetría del grupo espacial. Las posiciones generales de Wyckoff son aquellas invariantes únicamente ante la operación de identidad.

⁴La multiplicidad de una posición de Wyckoff indica el número de puntos equivalentes que se generan al aplicar todos los elementos de simetría del grupo espacial.

molécula al elegir un desplazamiento aleatorio en cada dirección de la celda, en coordenadas fraccionales.

- En el paso anterior se posiciona la unidad asimétrica. Así, se aplican los elementos de simetría del grupo espacial sobre la molécula posicionada de manera aleatoria para generar el resto de las moléculas, según el número de moléculas por celda indicado.
- Se verifica que no exista traslape entre los centros geométricos de las moléculas ni entre los átomos. Si la estructura cumple con la validación anterior entonces se conserva.
- Finalmente, al iterar sobre todos los grupos espaciales compatibles y probar diferentes posiciones aleatorias para cada grupo de simetría, se genera un conjunto vasto de estructuras cristalinas. El número de iteraciones y de estructuras generadas por cada grupo espacial están limitados, lo cual se especifica en el archivo de entrada.

Cabe enfatizar que existen dos datos de entrada del programa *Genarris* que no es posible conocer *a priori*: el volumen promedio y el número de moléculas de las celdas unitarias; de tal manera que sería necesario probar diferentes posibilidades. En el caso del volumen promedio, se ha propuesto solucionar esta indeterminación, en la versión más reciente del programa *Genarris* (Tom et al. 2020), al incluir un algoritmo de aprendizaje automático que ha sido entrenado con la base de datos estructural de Cambridge (CSD). Este algoritmo estima el volumen promedio de acuerdo con la molécula y el número de estas en la celda unitaria. Sin embargo, en los sistemas de referencia, al final se obtiene un volumen cercano al experimental (Tom et al. 2020); así, se evitó usar en este trabajo el algoritmo de predicción de volumen y se tomó directamente el volumen experimental de los sistemas considerados. Por otra parte, en el caso del número de moléculas en la celda unitaria, sigue siendo necesario considerar varias posibilidades. Sin embargo, la optimización de estos parámetros (volumen promedio y número de moléculas en la celda unitaria) no es parte de los objetivos de este trabajo y, por lo tanto, fueron tomados a partir de datos experimentales. Aunque no se descarta el análisis de estos parámetros una vez establecida la metodología de análisis configuracional.

Asimismo, resulta importante remarcar que el programa *Genarris* no considera las interacciones químicas dentro de las estructuras cristalinas generadas. Se toman en cuenta únicamente estructuras que son “espacialmente factibles”; es decir, sus moléculas no se traslapan. Pero no existe un criterio energético o químico que

favorezca la generación de algunas estructuras sobre otras. Sin embargo, esto no representa ningún inconveniente para el desarrollo de la metodología en esta tesis, ya que esta se plantea como un análisis previo a considerar la energía libre de las estructuras generadas e identificar las más estables. En consecuencia, se busca únicamente generar conjuntos de estructuras cristalinas cuyas configuraciones moleculares sean diferentes, con las cuales poder aplicar y validar la metodología de análisis configuracional descrita más adelante.

3. Hipótesis

Los cuaterniones duales pueden emplearse para describir y comparar las configuraciones de moléculas rígidas en redes cristalinas, tal que sea factible establecer conjuntos de estructuras cristalinas con configuraciones similares con métodos de agrupamiento, lo que contribuiría a identificar estructuras cristalinas relevantes (con configuraciones similares a las de un gran conjunto de estructuras) para la predicción de estructuras cristalinas.

4. Objetivos general y particulares

En esta tesis se busca desarrollar una metodología para la identificación de estructuras cristalinas moleculares con configuraciones similares. Con el fin de lograr lo anterior se plantean los siguientes objetivos particulares:

- Generar de manera aleatoria con el programa *Genarris* un conjunto inicial de estructuras cristalinas para tres compuestos moleculares: formamida, hidrazida maleica y ácido oxálico.
- Realizar una subdivisión en clústeres de los conjuntos iniciales de estructuras cristalinas en términos de sus energías totales obtenidas con cálculos de estructura electrónica.
- Desarrollar un algoritmo que permita describir y comparar las estructuras cristalinas en los conjuntos de cada compuesto en términos de las posiciones y orientaciones, en relación a una molécula de referencia, de las moléculas que constituyen las estructuras. Esto se hará implementando cuaterniones duales.
- Conjuntar el algoritmo desarrollado para describir y comparar la configuración espacial de estructuras cristalinas con métodos de agrupamiento con el fin de identificar clústeres de estructuras con configuraciones similares.

5. Metodología

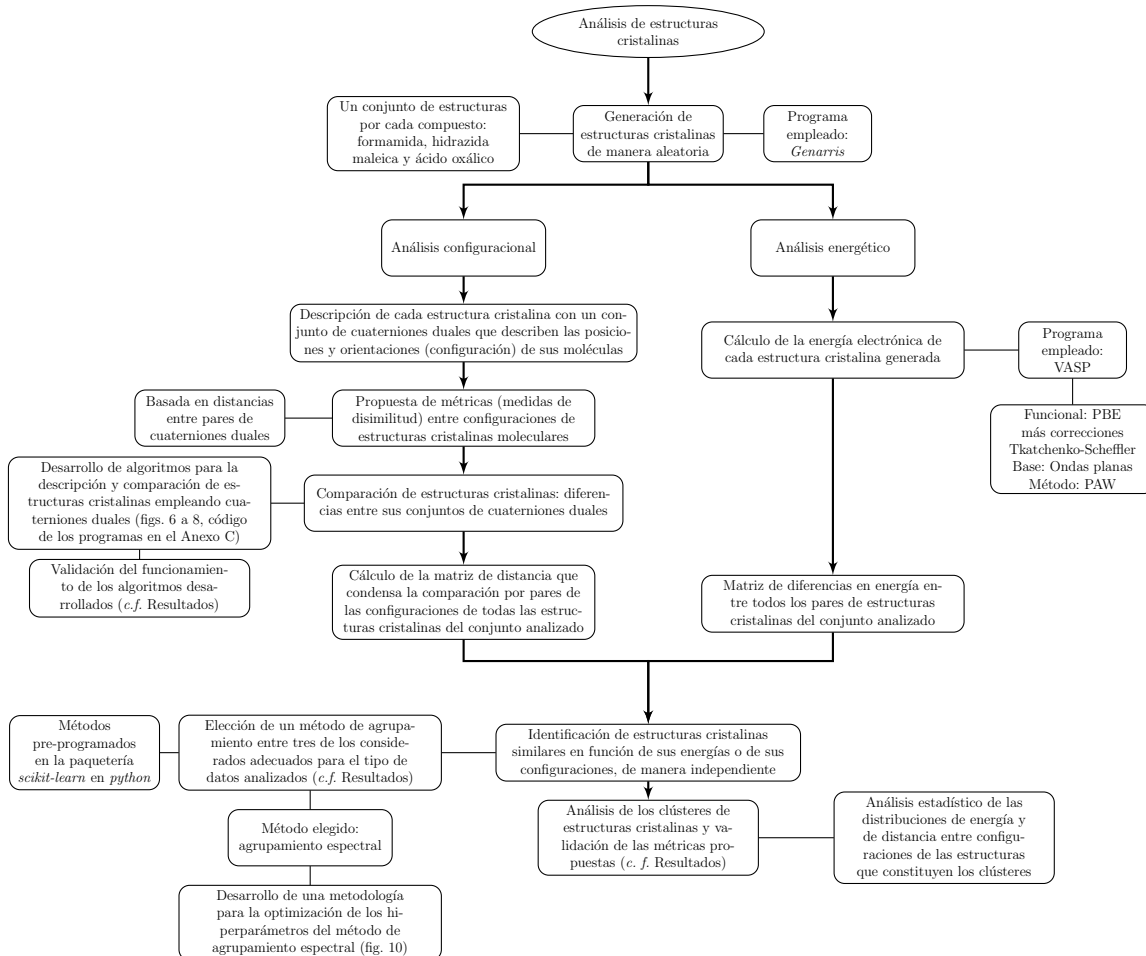


Figura 4: Esquema general de la metodología desarrollada en esta tesis. Las líneas rectas indican información sobre el trabajo adicional realizado para cumplir con lo planteado en el esquema principal señalado con flechas. Los detalles sobre cada punto planteado en el esquema se presentan en las siguientes secciones.

De manera general, como se presenta en la figura 4, se generan conjuntos de estructuras cristalinas con el programa *Genarris* (Li et al. 2018) para la formamida, la hidrazida maleica y el ácido oxálico. Para cada conjunto se realiza, por una parte, un análisis energético y, por otra parte, un análisis configuracional. En ambos casos, se aplican métodos de agrupamiento con el fin de identificar estructuras cristalinas semejantes, ya sea en términos de sus energías totales, obtenidas con métodos de estructura electrónica, o de la configuración de las moléculas en las redes cristalinas, con base en la metodología desarrollada en este proyecto. En las siguientes secciones se presentan los detalles de cada análisis.

Para el desarrollo de la metodología para el análisis de la configuración de moléculas en estructuras cristalinas se emplean cuaterniones duales. Por una parte, se describe el ambiente químico alrededor de la celda unitaria para cada estructura

cristalina de un conjunto de estructuras que se generan de manera aleatoria para un compuesto. Para esto se aplica un número finito de traslaciones periódicas de la celda unitaria, lo cual genera un conjunto de N moléculas y de este conjunto se selecciona una molécula de referencia. Las orientaciones y las posiciones de los centros geométricos del resto de las $M = N - 1$ moléculas en relación a la molécula de referencia son representadas con cuaterniones duales. Así, la configuración de las moléculas en cada red cristalina del conjunto generado aleatoriamente para un cierto compuesto es descrita con un conjunto de M cuaterniones duales.

Posteriormente, se comparan por pares las configuraciones de las estructuras cristalinas en el conjunto analizado a través de las diferencias entre los conjuntos de cuaterniones duales que las definen. Para comparar dos estructuras cristalinas es necesario medir la distancia entre todos los pares de cuaterniones duales de las dos estructuras. Debido a que, hasta donde sabemos, no existe una métrica que permita calcular la distancia entre cuaterniones duales, se proponen dos métricas que se denominan s y z . La comparación de todos los pares de cuaterniones duales equivale a comparar la orientación, en relación a la molécula de referencia, de todos los pares de moléculas del par de estructuras. En consecuencia, para comparar las configuraciones de un par de estructuras cristalinas se genera una matriz de comparación, $\mathbf{C}$.

Con el fin de simplificar la comparación de la configuración de estructuras cristalinas, la matriz $\mathbf{C}$ es reducida a un valor de distancia. Para este procedimiento se desarrollaron dos algoritmos, denominados *bmp* y *bmm*, que con base en la búsqueda de los valores más pequeños en la matriz $\mathbf{C}$ permiten “transformar” la comparación por pares de moléculas o pares de cuaterniones duales (matriz $\mathbf{C}$) a un valor de distancia que mide la diferencia entre las configuraciones del par de estructuras cristalinas. Al aplicar este procedimiento para todos los pares de estructuras cristalinas del conjunto analizado, se obtiene una matriz, $\mathbf{D}$, con los valores de la distancia entre las configuraciones de cada par de estructuras. La matriz de distancia, $\mathbf{D}$, depende de diferentes parámetros, los cuales se describen más adelante. Por lo tanto, para los conjuntos de estructuras cristalinas que se generan aleatoriamente para cada compuesto se calculan matrices de distancia con diferentes combinaciones de los parámetros que las definen.

Asimismo, siguiendo la metodología descrita en los párrafos anteriores se calculan las matrices de distancia empleando el error cuadrático medio (*RMSD*) y el descriptor de coordenadas relativo (*RCD*) entre las coordenadas de pares de moléculas en relación a la molécula de referencia. Estas métricas se emplean como puntos de comparación, ya que estas han sido exitosas en la comparación de la configuración de moléculas en redes cristalinas, y se aplican de acuerdo con las definiciones dadas en

el marco teórico (*cf.* secciones 2.4.2 y 2.4.3). Sin embargo, estas no se mencionan en el diagrama de flujo de la figura 4, ya que este se enfoca en las métricas estudiadas en este proyecto (s y z).

Se realiza también la comparación de las estructuras cristalinas de los conjuntos generados para cada compuesto en función de sus energías electrónicas. Para todas las estructuras generadas aleatoriamente se calculan sus energías totales con métodos estándar de estructura electrónica, cuyos detalles se encuentran en la sección 5.2. Cada par de estructuras se compara calculando la diferencia absoluta entre sus energías totales: $\Delta E = |E^\alpha - E^\beta|$, para un par de estructuras α y β . Tal que al comparar todos los pares de estructuras de un conjunto analizado se genera una matriz de distancia, $\mathbf{D}^{\Delta E}$.

Las matrices de distancia de energía o de la comparación de la configuración de moléculas en redes cristalinas se emplean para identificar estructuras similares con métodos de agrupamiento. Para esto se aplicaron tres metodologías diferentes: de OPTICS, de propagación de afinidades y espectral, entre las cuales se seleccionó el método de agrupamiento espectral como el adecuado para el análisis de las matrices de distancia empleadas, lo cual se presenta en la sección de resultados. En consecuencia, se desarrolló un algoritmo para la optimización de los hiperparámetros del método espectral, con el fin de generar clústeres de estructuras cristalinas de manera automática. Estos clústeres son el resultado de agrupar las estructuras cristalinas del conjunto completo de estructuras generadas aleatoriamente para un cierto compuesto en función de la similitud entre sus configuraciones o entre sus energías, según la matriz de distancia empleada. Se calculan diferentes matrices de distancia para generar los clústeres y se determina aquella que genere los clústeres óptimos, con base en una medida propia de los métodos de agrupamiento.

Finalmente, al aplicar la metodología descrita en esta sección a conjuntos de estructuras cristalinas generadas aleatoriamente para la formamida, la hidrazida maleica y el ácido oxálico, se analizó la relación entre los clústeres generados y la similitud entre las configuraciones de las estructuras cristalinas. Es decir, se estudió en qué extensión las matrices de distancia en conjunto con el método de agrupamiento espectral permiten capturar las similitudes configuracionales entre las estructuras cristalinas a través de los clústeres. Para esto se consideraron diferentes medidas estadísticas que se describen en la sección 5.6. Asimismo, se analizó la capacidad de las métricas para identificar polimorfos al aplicarlas para comparar las estructuras cristalinas identificadas experimentalmente para cada compuesto entre sí y con las generadas aleatoriamente para los mismos compuestos. De esta forma se validaron las diferentes métricas propuestas en esta tesis.

5.1. Generación del conjunto inicial de estructuras cristalinas

En una primera etapa, se generaron diferentes conjuntos de estructuras cristalinas con el programa *Genarris* (Li et al. 2018) para la formamida y por cada polimorfo del ácido oxálico ($OA-\alpha$ y $OA-\beta$) y de la hidrazida maleica ($MH1$, $MH2$ y $MH3$). Para cada caso se consideró como volumen promedio de las celdas unitarias a generar ($\bar{V}$) el volumen experimental de cada estructura (tab. 1) y como desviación estándar (σ_V) la mitad de este último ($\sigma_V = \bar{V}/2$). De igual manera, se mantuvo el número de moléculas por celda unitaria reportado en los datos experimentales (tab. 1). Las geometrías moleculares se extrajeron de los archivos *cif* de la *CSD* (Groom et al. (2016), ver tab. 1). En el caso del ácido oxálico se tomó la geometría del polimorfo $OA-\alpha$ y en el caso de la hidrazida maleica del polimorfo $MH1$, las cuales se usaron para la generación de las estructuras asociadas a cada compuesto sin hacer distinción en la geometría molecular en cada polimorfo. En el caso de OA y MH , las estructuras generadas por cada polimorfo fueron conjuntadas de manera aleatoria; así, se generó al final un único conjunto de estructuras cristalinas para cada compuesto.

Al generar las estructuras cristalinas con el programa *Genarris*, se buscó que el conjunto de estas fuese lo más vasto y representativo posible, dentro de los límites impuestos con los datos de la estructura experimental considerada: volumen (V) y número de moléculas (Z) de la celda unitaria. Para esto se modificó la cantidad máxima de estructuras por grupo espacial (entre los compatibles con el número de moléculas en la celda unitaria, *cf.* sección 2.6) sin cambiar el valor del resto de los parámetros ($\bar{V}$, σ_V y Z). Dado que el volumen de las celdas unitarias generadas se mantiene dentro del intervalo $[\bar{V} - \bar{V}/2, \bar{V} + \bar{V}/2]$, el número de estructuras cristalinas cuyas moléculas no se traslapan puede ser pequeño para cada grupo espacial (compatible con el valor de Z). Sin embargo, la cantidad de estructuras generadas también puede estar afectada por las restricciones en el número máximo de estructuras cristalinas o por el número máximo de iteraciones, por grupo espacial. Así, se aseguró que los valores de estos dos últimos parámetros fuesen lo suficientemente grandes, tal que las estructuras cristalinas generadas por cada grupo espacial fuesen representativas. Es decir, estructuras cristalinas que fuesen las “aceptables” (cuyas moléculas no se traslapan) entre un gran número de posibles estructuras, el cual a su vez no estuviese limitado por las restricciones previamente mencionadas (número máximo de estructuras y de iteraciones por grupo espacial). En el caso del número máximo de iteraciones, se mantuvo el valor estándar de 2000000. Mientras que el número máximo de estructuras por grupo espacial se varió hasta 8000 y se buscó que

el número total de estructuras generadas no cambiase demasiado al modificar este parámetro. Por cada estructura cristalina determinada experimentalmente (OA- α , OA- β , *MH1*, *MH2*, *MH3* y formamida- α), se seleccionó un único valor del máximo de estructuras por grupo espacial en el rango de valores para el cual no se observaron cambios importantes del número total de estructuras cristalinas. Sin embargo, cabe mencionar que la elección del parámetro previamente mencionado sigue siendo en cierta medida arbitraria debido a la aleatoriedad del programa *Genarris* (*cf.* sección 2.6). Aun así, se observó cierta convergencia en el número total de estructuras generadas respecto al número máximo de estructuras cristalinas por grupo espacial. Tal que no fue posible obtener conjuntos de estructuras cristalinas de tamaños considerablemente mayores a los que se reportan en los resultados. Solo en el caso de la formamida no se observó un límite en el número total de estructuras cristalinas y, por lo tanto, este fue elegido de manera arbitraria con el único fin de analizar un conjunto amplio de estructuras cristalinas. Los valores de los parámetros empleados se muestran en la tabla 2.

Referencia	N_{max}	$\bar{V}$	σ_V	Z	N_s
Formamida	1000	230.45	115.23	4	15594
OA- α	1000	307.02	153.51	4	743
OA- β	1000	157.01	78.51	2	227
<i>MH1</i>	2000	229.4	114.7	2	122
<i>MH2</i>	1000	455.9	227.95	4	449
<i>MH3</i>	2000	466.68	233.34	4	521

Tabla 2: Valores del número máximo de estructuras por grupo espacial (compatible con Z), N_{max} , del volumen promedio, $\bar{V}$, de la desviación estándar del volumen ($\sigma_V = \bar{V}/2$) y del número de moléculas, Z , de las celdas unitarias que fueron empleados para generar N_s estructuras cristalinas aleatoriamente con el programa *Genarris* por cada estructura cristalina conocida experimentalmente para los compuestos analizados en esta tesis. El volumen promedio y el número de moléculas en la celda unitaria se tomaron de los datos experimentales reportados en la tabla 1. Las estructuras cristalinas generadas por cada polimorfo del ácido oxálico y de la hidrazida maleica fueron conjuntadas de manera aleatoria, tal que el número total de estructuras analizadas fue de 970 y 1092, respectivamente.

5.2. Análisis energético de las estructuras cristalinas

Para cada una de las estructuras generadas con *Genarris* se calcula la energía electrónica con el programa VASP (Kresse y Hafner 1993, Kresse y Furthmüller 1996, Kresse y Furthmüller 1996), empleando el funcional PBE en conjunto con la corrección Tkatchenko-Scheffler (Perdew et al. 1996, Bučko et al. 2013) y el método PAW (Kresse y Joubert 1999b) (*cf.* sección 2.2.3). El funcional PBE y la corrección de interacciones de dispersión de London se seleccionaron ya que son metodologías que en conjunto generan buenos resultados en cálculos de estructura electrónica de cristales moleculares (Rappoport et al. 2009, Hasnip et al. 2014), de acuerdo con lo presentado en los antecedentes (*cf.* sección 2.2.2). Todos los cálculos se realizaron en el clúster de supercómputo YOLTLA de la UAM-Unidad Iztapalapa en un total de 40 procesadores Intel® Xeon® E5 v3.

En una primera etapa, de acuerdo con la metodología PAW (*cf.* sección 2.2.3), es necesario hacer un análisis de convergencia de la energía total de una de las estructuras del conjunto analizado en relación al número de ondas planas y al número de puntos $\vec{k}$. Por una parte, se busca el límite de la suma de la combinación lineal de ondas planas (ecs. 35 y 36), también conocido como un corte de energía (*cf.* sección 5.3.1, ec. 38), a partir del cual la energía total se mantiene constante (variaciones menores a 0.1 kcal/mol). Por otra parte, manteniendo el valor óptimo del corte en energía, se busca el número de puntos $\vec{k}$ necesarios para calcular las integrales de las cuales depende la energía total en el espacio recíproco (*cf.* sección 5.3.1, ec. 39), tal que para el número de puntos $\vec{k}$ elegido igual se llegue a la convergencia en la energía. Una vez establecido lo anterior para una estructura del conjunto analizado, elegida de manera aleatoria, se mantiene el valor óptimo del corte en energía para el resto de las estructuras. En el caso de los puntos $\vec{k}$, dado que su número depende del tamaño de la celda unitaria, es necesario corregir los valores para cada estructura de acuerdo con la proporción entre las longitudes de los ejes que definen su celda unitaria y los correspondientes a la estructura con la que se analizó la convergencia respecto a este parámetro. Tal que

$$k_i^\alpha = \left\lceil \frac{\alpha_0}{\alpha_i} k_0^\alpha \right\rceil \quad (92)$$

donde k_0^α es el número de puntos $\vec{k}$ en la dirección α ($\alpha=\vec{a}$, $\vec{b}$ o $\vec{c}$) de la celda unitaria considerada en el análisis de convergencia, α_0 es el parámetro de red en la misma dirección; k_i^α y α_i son los parámetros correspondientes de cualquier otra estructura en el conjunto analizado; $\lceil x \rceil$ es la función techo que asigna el número entero más cercano al número real evaluado. Tal que, por ejemplo, si en alguna de

las direcciones el parámetro de red de la estructura i es la mitad del de la referencia, entonces se toma el doble de puntos $\vec{k}$ en esa dirección, tal que se mantiene la convergencia de la energía respecto al número de puntos $\vec{k}$. Así, una vez determinado el número de puntos $\vec{k}$ para cada sistema, se procede al cálculo de la energía electrónica de todas las estructuras cristalinas.

5.3. Análisis configuracional con cuaterniones duales

5.3.1. Análisis de coordenadas de sistemas periódicos

Con el fin de generar los cuaterniones y cuaterniones duales necesarios para el estudio configuracional de las estructuras cristalinas, fue necesario desarrollar un algoritmo para el tratamiento de las coordenadas de cada sistema dadas las condiciones periódicas. Así, en una primera parte se realizó la verificación de las coordenadas de los átomos de las moléculas en la celda central. Debido a la periodicidad de los sistemas, podría ser que las coordenadas de los átomos de ciertas moléculas en la celda central no pertenecieran a estas sino a alguna de sus traslaciones periódicas. Por lo tanto, se realizó para cada molécula en la celda central una búsqueda de los vectores más pequeños entre uno de sus átomos y el resto. Así, se construyeron los siguientes vectores para considerar todas las direcciones en las que podrían generarse traslaciones periódicas de las moléculas centrales:

$$\begin{aligned} \vec{v}_1 &= \vec{a} & \vec{v}_4 &= \vec{a} + \vec{b} & \vec{v}_7 &= \vec{a} - \vec{b} & \vec{v}_{10} &= \vec{a} + \vec{b} + \vec{c} & \vec{v}_{13} &= \vec{a} - \vec{b} - \vec{c} \\ \vec{v}_2 &= \vec{b} & \vec{v}_5 &= \vec{a} + \vec{c} & \vec{v}_8 &= \vec{a} - \vec{c} & \vec{v}_{11} &= \vec{a} - \vec{b} + \vec{c} \\ \vec{v}_3 &= \vec{c} & \vec{v}_6 &= \vec{b} + \vec{c} & \vec{v}_9 &= \vec{b} - \vec{c} & \vec{v}_{12} &= \vec{a} + \vec{b} - \vec{c} \end{aligned} \quad (93)$$

donde $\vec{a}$, $\vec{b}$ y $\vec{c}$ son los vectores que describen a la celda unitaria. De acuerdo con lo anterior, para un vector $\vec{w}_j$ dado entre un átomo j y un átomo de referencia en una molécula central, tomado siempre de manera arbitraria como el primero en la lista de átomos de la molécula, se probó la siguiente condición:

$$\vec{w}_j' = \vec{w}_j - \text{nint}\left(\frac{\vec{w}_i \cdot \vec{v}_i}{|\vec{v}_i|}\right)\vec{v}_i \quad (94)$$

donde nint implica que el valor real evaluado es redondeado al número entero más cercano. Así, si la proyección del vector $\vec{w}_j$ sobre alguna de las direcciones de la celda unitaria, $\vec{v}_i$, es mayor a la mitad de la magnitud de este último ($\vec{w}_i \cdot \vec{v}_i > \frac{1}{2}|\vec{v}_i|$), $\vec{w}_j$ se modifica de tal manera que la proyección del vector resultante se encuentre

dentro del intervalo mencionado, $(0, \frac{1}{2}|\vec{v}_i|)$. De esta manera, al probar la condición anterior con cada una de las direcciones, se obtienen los vectores más pequeños posibles entre cada átomo y el átomo de referencia de la molécula analizada. Al aplicar este procedimiento a cada una de las moléculas centrales se obtienen las coordenadas de sus átomos correctas.

Posteriormente, se calculan los centros geométricos de las moléculas centrales a partir de las coordenadas de sus átomos corregidas:

$$c\vec{g}_i = \frac{1}{n} \sum_{j=1}^n \vec{x}_j^i \quad (95)$$

donde $c\vec{g}_i$ son las coordenadas del centro geométrico de la molécula i , n es el número de átomos en la molécula y $\vec{x}_j^i$ son las coordenadas del átomo j perteneciente a la molécula i .

Finalmente, se generan las traslaciones periódicas de la celda central con el fin de obtener el ambiente químico alrededor de esta. Para esto se emplea una variable z_p que toma valores enteros mayores o iguales a cero, la cual define el límite de las traslaciones periódicas alrededor de la celda unitaria central. En otras palabras, se consideran desplazamientos de esta última en máximo z_p unidades en cada una de las direcciones del espacio posibles ($\vec{v}_i$, ec. 93), sea

$$\vec{w}' = \vec{w} + i\vec{a} + j\vec{b} + k\vec{c} \quad \forall \quad i, j, k \in \mathbb{Z} \in [-z_p, z_p] \quad (96)$$

donde $\vec{w}$ hace referencia tanto a las coordenadas de los átomos como de los centros geométricos de las moléculas centrales y $\vec{w}'$ son las traslaciones periódicas correspondientes. Así, en términos de la programación de este algoritmo, se tienen tres ciclos anidados (por cada uno de los vectores que definen la celda unitaria) que en conjunto generan en cada paso una traslación periódica de la celda central con sus moléculas. Esto resulta en $(2z_p + 1)^3$ traslaciones, ya que en cada dirección ($\vec{v}_i$, ec. 93) se consideran $2z_p$ desplazamientos de la celda central. Entonces, si en la celda central se tienen Z moléculas, se genera un total de

$$N = (2z_p + 1)^3 Z \quad (97)$$

moléculas. Sin embargo, como se describe más adelante, solo se calculan los cuaterniones duales de $M = N - 1$ moléculas, ya que se selecciona una molécula del conjunto de N moléculas en relación a la cual se realiza el cálculo de los cuaternio-

nes duales. De esta forma, el ambiente químico de las estructuras cristalinas queda representado por M moléculas alrededor de una molécula de referencia.

5.3.2. Caso particular: estructuras con diferente número de moléculas en la celda unitaria

Al analizar un conjunto de estructuras en el cual el número de moléculas en la celda unitaria varía, es necesario generar un número diferente de traslaciones periódicas en cada caso. Esto se debe a que la metodología propuesta en este proyecto plantea la comparación de un número fijo (l) de moléculas alrededor de la molécula de referencia para cada par de estructuras, como se describirá más adelante. Por lo tanto, es necesario asegurar que se obtengan para cada estructura al menos l moléculas al definir el ambiente químico en la red cristalina (*cf.* sección 5.3.1). En otras palabras, se verifica que $M \geq l$ al calcular el valor de $M = (2z_p + 1)^3 Z - 1$ con el límite z_p y el número de moléculas en la celda unitaria, Z , dados en el archivo de entrada del programa desarrollado en este proyecto. Si la condición $M \geq l$ no se cumple, se aumenta en una unidad el límite z_p hasta cumplir con la condición mencionada. Una vez concluido este proceso, el mayor valor de z_p será el límite considerado al aplicar las traslaciones periódicas de la celda central y generar el número total de moléculas (N) de la estructura cristalina. Este procedimiento se aplica para cada estructura en el conjunto generado con el programa *Genarris* para el compuesto analizado. Evidentemente, en el caso en que todas las estructuras tengan el mismo número de moléculas en la celda unitaria, el número total de moléculas (N) será el mismo para todo el conjunto, como es el caso de la formamida. Este paso de la metodología se resume en el diagrama de flujo presentado en la figura 5.

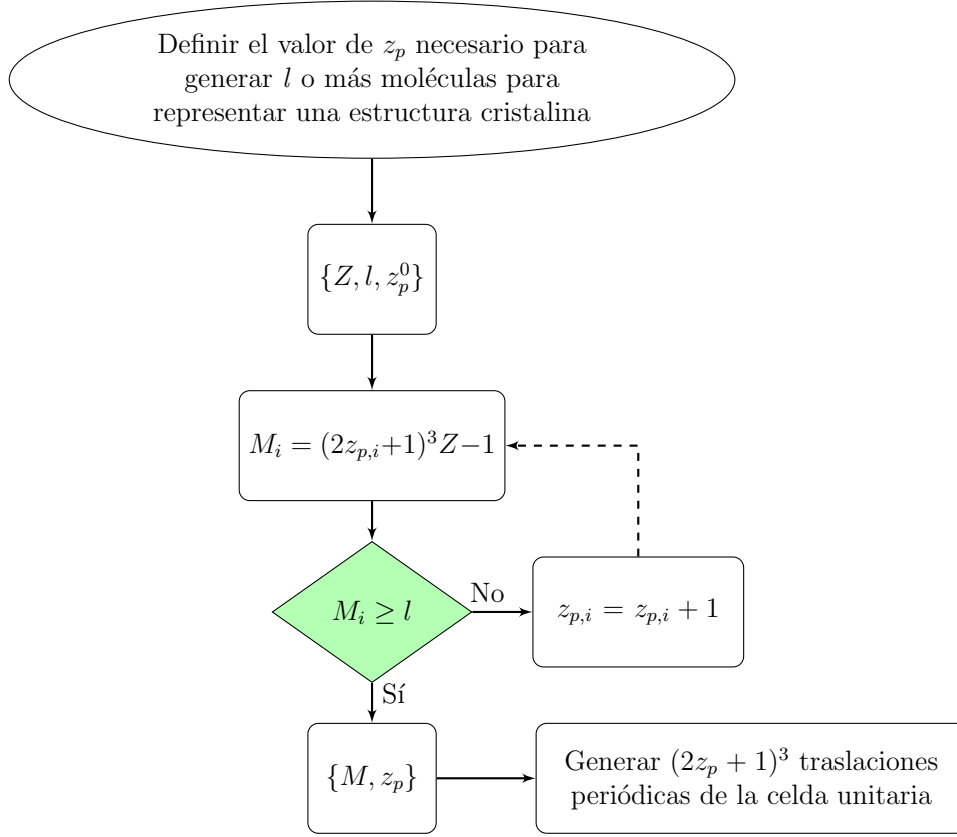


Figura 5: Diagrama de flujo para definir el límite de traslaciones periódicas de la celda central, z_p , tal que el número de moléculas que representen una estructura cristalina, M , generadas al trasladar periódicamente la celda unitaria, sea mayor o igual al número de moléculas a comparar entre cada par de estructuras, l .

5.3.3. Generación de cuaterniones duales

Para generar los cuaterniones duales para cada estructura generada con el programa *Genarris* se parte del conjunto de coordenadas establecidas para los átomos en las N moléculas, como se describió en la sección anterior. Se selecciona una molécula de la celda central como referencia y se calculan los cuaterniones duales de las M moléculas restantes ($M = N - 1$), de acuerdo con lo descrito en los antecedentes (cf. sección 2.4.1). Así, para cada estructura cristalina se obtiene un conjunto de M cuaterniones duales, tal que M puede ser diferente para cada estructura en el conjunto analizado si estas difieren en el número de moléculas en la celda unitaria (fig. 5). El esquema general del cálculo de los cuaterniones duales se presenta en el diagrama de flujo de la figura 6.

Con el fin de asegurar el funcionamiento correcto del algoritmo desarrollado, cada cuaternión dual fue verificado al aplicarlo sobre cada átomo de la molécula de referencia y restar las coordenadas resultantes de las coordenadas del átomo equivalente en la molécula asociada al cuaternión dual en cuestión. Esto de acuerdo con el producto definido en la ecuación 84 (cf. sección 2.4.1). Esto último no forma parte del análisis configuracional de las estructuras cristalinas moleculares.

Por otra parte, de acuerdo con la metodología descrita en la siguiente sección, la comparación de dos estructuras cristalinas de un mismo compuesto molecular implica la búsqueda de los l valores más pequeños en una matriz de comparación, $\mathbf{C}$, que contiene las distancias entre todos los pares de cuaterniones duales de las dos estructuras. Para analizar esta matriz se desarrolló un algoritmo denominado algoritmo de búsqueda del máximo mínimo (*bmm*), descrito más adelante. Este algoritmo requiere que el número de cuaterniones duales generados (M) sea igual al valor de l para todas las estructuras que se vayan a comparar. Esto se logra al reducir el conjunto de M cuaterniones duales a los l cuaterniones duales con la menor distancia a la molécula de referencia, según las magnitudes de sus vectores $\vec{T}$ correspondientes, para cada estructura del conjunto analizado. De este modo, aunque en un inicio se generen M cuaterniones duales (fig. 6) para cada estructura cristalina, este conjunto será reducido a l cuaterniones duales, solo en el caso del algoritmo *bmm* debido al tipo de comparación que este implica.

Los cuaterniones duales generados para una estructura cristalina siempre serán referidos a una de las moléculas que constituyen la celda unitaria central, para lo cual es elegida la primera molécula en la lista de moléculas centrales. Así, cada estructura cristalina, en el conjunto de estructuras para cada compuesto, es representada por un conjunto de cuaterniones duales asociados a las moléculas que rodean

una molécula que se seleccionó como punto de referencia (representadas con triadas de vectores ortonormales y sus centros geométricos). Entonces, con el fin de poder comparar los conjuntos de cuaterniones duales obtenidos en relación a las moléculas de referencia en las diferentes estructuras cristalinas generadas de manera aleatoria, se calculó adicionalmente un conjunto de cuaterniones duales entre las moléculas de referencia. En otras palabras, para poder comparar un par de estructuras cristalinas fue necesario calcular la transformación que lleva de la molécula de referencia de una de las estructuras a la molécula de referencia de la otra. Dado que las moléculas de referencia siempre se colocan en el origen de las estructuras cristalinas, estas solo varían de una estructura a otra con respecto a su orientación. Por lo tanto, considerando dos redes cristalinas moleculares α y β , la transformación que lleva de una molécula de referencia a otra involucra la rotación:

$$\sigma_{\alpha\beta}^{ref} = q_{\alpha\beta}^R + \epsilon \mathbf{0} \quad (98)$$

Tal que se cumple

$$\sigma_{\beta}^{ref} = \sigma_{\alpha\beta}^{ref} \sigma_{\alpha}^{ref} \sigma_{\alpha\beta}^{ref\circ} \quad (99)$$

donde los subíndices detonan el sistema al que pertenece cada cuaternión dual. El método utilizado para obtener los cuaterniones $q_{\alpha\beta}^R$ es el descrito en los antecedentes (ecs. 63 y 80 a 82). Una vez calculados los cuaterniones duales para todas las estructuras de un compuesto, se calcularon los cuaterniones entre moléculas de referencia al iterar sobre todos los pares de estructuras cristalinas posibles, tal que al final se genera el siguiente conjunto de cuaterniones duales:

$$Q_{ref} = \{\sigma_{1,1}, \sigma_{1,2}, \dots, \sigma_{\alpha,\alpha}, \sigma_{\alpha,\alpha+1}, \dots, \sigma_{N_s,N_s}\} \quad (100)$$

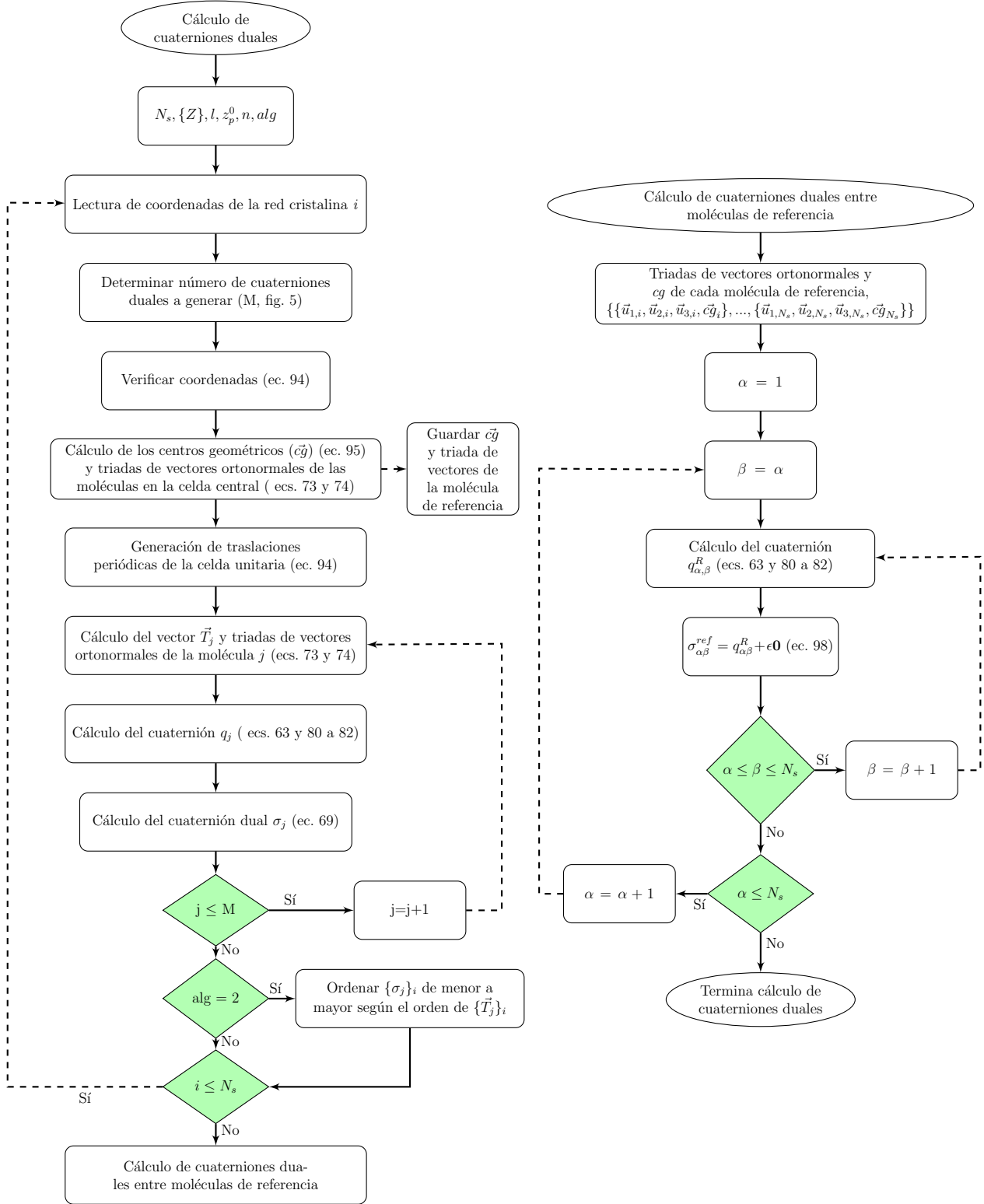


Figura 6: Diagrama de flujo del cálculo de cuaterniones duales para un conjunto de estructuras cristalinas de un compuesto químico molecular. Sea *alg* un parámetro que determina el tipo de algoritmo de búsqueda de mínimos a emplear (*bmp* o *bmm*, descritos en la siguiente sección).

5.4. Comparación de las configuraciones y energías de estructuras cristalinas moleculares

Con el fin de agrupar o generar clústeres de estructuras cristalinas con configuraciones similares, se implementaron tres métricas diferentes. El objetivo de esto fue obtener al final matrices cuyas entradas tuviesen el valor de la distancia, en términos de la similitud entre los arreglos de las moléculas en las redes cristalinas (configuración), entre pares de estructuras. Estas matrices son la información de entrada de los métodos de agrupamiento. De acuerdo con lo mencionado en el marco teórico (*cf.* sección 2.3), las métricas deben tomar valores grandes para estructuras diferentes, ser cero para la misma estructura ($d_{ii} = 0$) y simétricas ($d_{ij} = d_{ji}$). Estas propiedades fueron la principal guía en la implementación de las tres métricas descritas a continuación.

De acuerdo con lo descrito en la sección anterior, cada estructura cristalina molecular es representada por un conjunto de M cuaterniones duales referidos a una molécula del sistema. Así, es necesario homogeneizar los marcos de referencia de los cuaterniones duales al comparar las estructuras cristalinas por pares. Por lo tanto, los cuaterniones duales de una estructura β son transformados con base en la orientación de la molécula de referencia de una estructura α de acuerdo con la siguiente expresión:

$$\sigma_j^\alpha = \sigma_{\beta\alpha}^{ref} \sigma_j^\beta \sigma_{\beta\alpha}^{ref*} \quad (101)$$

donde $\sigma_{\beta\alpha}^{ref}$ es el cuaternión dual que rota la molécula de referencia en la estructura β a la orientación de la molécula de referencia en la estructura α (ecs. 98 y 99), σ_j^β es el cuaternión dual que representa la posición y orientación de la molécula j en la estructura β en relación a la molécula de referencia en la misma estructura y σ_j^α es el mismo cuaternión dual, pero en relación a la molécula de referencia en la estructura α . De esta manera, la comparación de los cuaterniones duales del par de estructuras a analizar se realiza de acuerdo con el siguiente producto:

$$\sigma_i^\alpha \otimes \sigma_j^{\alpha*} = \sigma_i^\alpha \otimes (\sigma_{\alpha\beta}^* \sigma_j^\beta \sigma_{\alpha\beta})^* \quad (102)$$

donde los cuaterniones duales σ_i^α y σ_j^α pertenecen a las estructuras α y β , respectivamente, pero están expresados en relación a la molécula de referencia de la estructura α . Si ambas estructuras son iguales, lo cual implica que sus referencias pueden estar rotadas entre sí, y los cuaterniones duales son los mismos (representan la posición y orientación de la misma molécula), se cumple lo siguiente:

$$\sigma_i^\alpha \otimes \sigma_j^{\alpha*} = (1, 0, 0, 0) + \epsilon(0, 0, 0, 0) \quad (103)$$

En el caso de que los cuaterniones duales difieran, las componentes del cuaternión dual resultante del producto $\sigma_i^\alpha \otimes \sigma_j^{\alpha*}$ serán diferentes de cero (ec. 104):

$$\sigma_i^\alpha \otimes \sigma_j^{\alpha*} = (r_{ij}, rx_{ij}, ry_{ij}, rz_{ij}) + \epsilon(s_{ij}, sx_{ij}, sy_{ij}, sz_{ij}) \quad (104)$$

En particular, de este producto (ec. 104) se analiza el valor absoluto de la parte escalar en la componente dual, denotada por simplicidad como s . De acuerdo con el producto de cuaterniones duales (Jia s.f.) y la ecuación 69, la componente dual en la ecuación 104 involucra el producto de los cuaterniones asociados con las traslaciones y las rotaciones; por lo tanto, es factible que esta componente mida la diferencia entre las posiciones y orientaciones, en relación a la molécula de referencia, de las moléculas asociadas a los cuaterniones duales comparados. Además, con el fin de también considerar la parte vectorial de la parte dual, se realiza el producto del cuaternión correspondiente (ec. 104) y su conjugado,

$$(s_{ij}, sx_{ij}, sy_{ij}, sz_{ij})(s_{ij}, -sx_{ij}, -sy_{ij}, -sz_{ij}) = (z_{ij}, 0, 0, 0) \quad (105)$$

cuyo parámetro resultante será referido como z . De acuerdo con lo mostrado en los antecedentes (*c.f.* Comparación de cuaterniones duales, sección 2.4.1), este tipo de operación (ec. 105) resulta en el cuadrado de la magnitud del cuaternión. En el caso de que los cuaterniones duales comparados fuesen idénticos, el cuaternión de la parte dual sería nulo y, por lo tanto, también lo serían los parámetros s y z . Así, s y z pueden servir como medidas de disimilitud o métricas; es decir, valores grandes de estos parámetros indican una mayor distancia (diferencia) entre los cuaterniones duales y, por lo tanto, entre las posiciones y orientaciones de las moléculas correspondientes. Por otra parte, dado que el parámetro z es el cuadrado de la magnitud del cuaternión considerado, también se empleó la raíz cuadrada de z , la cual será referida como z_{rc} .

Los parámetros s , z y z_{rc} fueron calculados para todos los productos posibles de cuaterniones duales entre cada par de estructuras, lo cual genera las matrices de comparación $\mathbf{S}$, $\mathbf{Z}$ y $\mathbf{Z}_{rc}$, respectivamente. Adicionalmente, se implementaron las métricas *RCD* (Li et al. 2018), de acuerdo con lo descrito en la sección 2.4.3, y el *RMSD* (*cf.* sección 2.4.2), cuyas matrices serán referidas con las mismas letras en negritas.

En el caso del *RMSD*, se emplearon los conjuntos de cuaterniones duales gene-

rados para cada estructura cristalina. Al comparar un par de estructuras cristalinas α y β , los cuaterniones duales de la estructura β son transformados de acuerdo con la ecuación 101; por lo tanto, los conjuntos de cuaterniones duales del par de estructuras α y β están referidos a la misma orientación de la molécula de referencia. Esto implica, que ambos sistemas se traslapan en términos de sus moléculas de referencia. Así, se pueden determinar las diferencias entre las coordenadas de las dos estructuras cristalinas. No existe manera de relacionar los índices de las moléculas de ambos sistemas; es decir, no existe manera de saber *a priori* qué molécula en la estructura α le es equivalente a una molécula en la estructura β , lo cual sí puede hacerse, por ejemplo, entre los átomos de dos moléculas del mismo compuesto. Por lo tanto, la comparación entre las coordenadas del par de estructuras α y β se hace entre todos los pares posibles de moléculas de acuerdo con la siguiente expresión:

$$\mathbf{RMSD}_{i,j} = \left(\frac{1}{N} \sum_{a=1}^n \left| \sigma_i^\alpha \otimes (1 + \epsilon v_a^{ref,\alpha}) \otimes \sigma_i^{\alpha\circ} - \sigma_j^\alpha \otimes (1 + \epsilon v_a^{ref,\alpha}) \otimes \sigma_j^{\alpha\circ} \right|^2 \right)^{1/2} \quad (106)$$

Esta ecuación implica que los cuaterniones duales asociados a las moléculas i (perteneciente a la red cristalina α) y j (perteneciente a la red cristalina β , ec. 101) son aplicados a cada átomo de la molécula de referencia en la estructura α . Así, ambas transformaciones $(\sigma_i^\alpha \otimes (1 + \epsilon v_a^{ref,\alpha}) \otimes \sigma_i^{\alpha\circ})$ y $(\sigma_j^\alpha \otimes (1 + \epsilon v_a^{ref,\alpha}) \otimes \sigma_j^{\alpha\circ})$ generan los átomos equivalentes en las moléculas correspondientes a los cuaterniones duales empleados. Por lo tanto, si se comparan dos estructuras cristalinas idénticas, el *RMSD* de las distancias entre las coordenadas de los átomos de las mismas moléculas es cero: $\mathbf{RMSD}_{i,i}=0$. Pero, conforme las configuraciones de los sistemas difieren, esta medida aumenta. Es decir, aumentan las distancias entre las coordenadas de los átomos de las dos estructuras cristalinas en relación a la molécula de referencia y, por lo tanto, incrementa la distancia entre las configuraciones del par de estructuras.

Así, para cada comparación de un par de estructuras cristalinas se generó una matriz de tamaño $M \times M'$, que incluye una métrica (z , z_{rc} , s , *RCD* o *RMSD*), tal que M es el número de cuaterniones duales de la estructura α y M' es el número de cuaterniones duales de la estructura β . Esto fue un primer paso en la comparación; sin embargo, fue necesario condensar aún más la información con el fin de obtener el valor de la distancia entre las configuraciones de cada par de estructuras cristalinas (α y β), en lugar de una matriz de comparación ($\mathbf{Z}$, $\mathbf{Z}_{rc}$, $\mathbf{S}$, *RCD* y *RMSD*), y generar así matrices de distancia, $\mathbf{D}$. Es decir, matrices que contienen el valor de la distancia entre cada par de estructuras cristalinas. Por lo tanto, es importante

remarcar que el término “matriz de comparación” hace referencia al cálculo de las distancias entre pares de moléculas (representadas con cuaterniones duales) de dos estructuras cristalinas, mientras que el término “matriz de distancia” hace referencia al cálculo de las distancias entre las configuraciones de todos los pares de estructuras cristalinas. La transformación de las matrices de comparación, $\mathbf{C}$, a valores de la distancia entre configuraciones cristalinas se realizó con dos algoritmos diferentes denominados búsqueda del mínimo promedio (*bmp*) y búsqueda del máximo mínimo (*bmm*), los cuales fueron desarrollados en este proyecto y se describen a continuación.

Entonces, se propone promediar directamente los l valores más pequeños de las matrices de comparación ($\mathbf{Z}$, $\mathbf{Z}_{rc}$, $\mathbf{S}$, $\mathbf{RCD}$ y $\mathbf{RMSD}$). Esto también equivale a comparar las l moléculas más cercanas (en posición y orientación) a la molécula de referencia en cada par de estructuras cristalinas. Este método se identificará como el algoritmo de búsqueda del mínimo promedio (*bmp*):

$$\mathbf{D}_{bmp,\alpha,\beta}^c = \frac{1}{l} \sum_{k=1}^l \min_k(\mathbf{C}^{\alpha,\beta}) \quad (107)$$

donde $\mathbf{C}$ es cualquiera de las matrices de comparación mencionadas previamente: $\mathbf{Z}$, $\mathbf{Z}_{rc}$, $\mathbf{S}$, $\mathbf{RCD}$ y $\mathbf{RMSD}$. Esta metodología se resume en la figura 7a. Cabe mencionar que en el caso de la medida *RCD* se omite el factor $1/l$; es decir, se suman directamente los valores más pequeños. Esto con el fin de mantener la propuesta de la medida dada en el artículo de Li et al. (2018) (ec. 91). Sin embargo, debido a que este solo es un factor de escalamiento, esto no tiene ningún efecto en las agrupaciones que se obtengan y, por lo tanto, se referirá al método *bmp* para esta medida sin hacer una distinción al respecto.

Con el fin de establecer mejores formas de comparar un par de estructuras cristalinas α y β , se propone adicionalmente un algoritmo que será referido como búsqueda del máximo mínimo (*bmm*). Este busca generar la comparación en términos de la mayor diferencia posible entre los pares de moléculas más cercanas a la molécula de referencia. En este caso, primero se ordenan los cuaterniones duales de todas las estructuras de menor a mayor distancia a las moléculas de referencia correspondientes ($|\vec{T}|$, cf. sección 2.4.1) y se toman los l cuaterniones duales más cercanos a la molécula de referencia de cada estructura, para calcular las matrices de comparación cuadradas con dimensión $l \times l$, a diferencia del método *bmp* para el cual las matrices pueden ser rectangulares ($M \times M'$). Así, el algoritmo busca los valores más pequeños por columna y por renglón en la matriz de comparación, de manera independiente. Considerando como ejemplo la búsqueda por renglón, en cada iteración se verifica si el nuevo mínimo se encuentra en la misma columna que alguno de los previamente

hallados. Si lo anterior se cumple, entonces se conserva el valor máximo entre los dos mínimos que se encuentran en la misma columna y se repite la búsqueda en la fila correspondiente al valor más pequeño. Esto asegura que los valores mínimos no coincidan ni en columna ni en renglón, tal que los valores de las matrices de comparación sean lo más representativos posible. Al final, de los mínimos encontrados por columna y por renglón se busca el valor máximo para cada caso y el mayor de estos dos se elige como el valor de distancia entre las configuraciones del par de estructuras α y β (fig. 7b y ec. 108).

$$\mathbf{D}_{bmm,\alpha,\beta}^c = \max(\min_{bmm}^{columna}(\mathbf{C}^{\alpha,\beta'}), \min_{bmm}^{renglón}(\mathbf{C}^{\alpha,\beta'})) \quad (108)$$

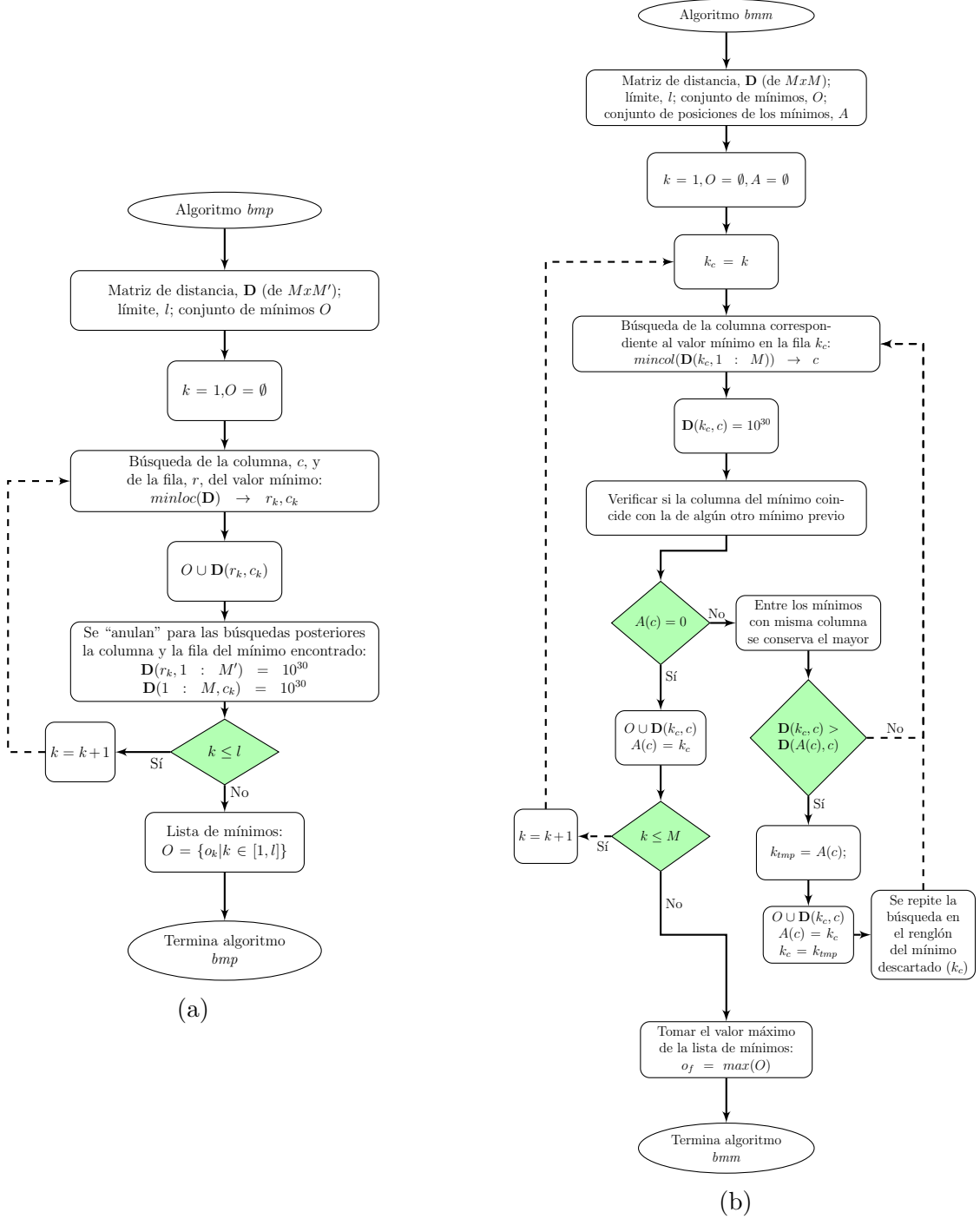


Figura 7: Diagramas de flujo mostrando los algoritmos de búsqueda de valores mínimos (a) *bmp* y (b) *bmm*. En el caso del algoritmo *bmm* se muestra solo el caso de la búsqueda por renglones; sin embargo, este algoritmo es equivalente a la búsqueda por columnas al invertir los índices de la matriz $\mathbf{D}$.

Por último, en el caso de las energías totales, las matrices de distancia se calcularon simplemente como el valor absoluto de la diferencia de energía en *kcal/mol* entre cada par de estructuras: $\Delta E_{\alpha\beta} = |E_{\alpha} - E_{\beta}|$, tal que estas serán referidas como $\mathbf{D}^{\Delta E}$. Esto es la única diferencia en esta parte de la metodología entre el análisis configuracional y el energético.

En resumen, para un par de estructuras cristalinas α y β se genera una matriz de comparación, $\mathbf{C}$, que representa una métrica (s , z , **RCD** o **RMSD**) entre las posiciones y orientaciones relativas de todos los pares de moléculas i (de la estructura α) y j (de la estructura β). A su vez se determinan los l valores más pequeños en la matriz de comparación de acuerdo con el método *bmp* o con *bmm* (ecs. 107 y 108), de tal manera que se obtiene un valor de la distancia entre las configuraciones del par de estructuras cristalinas α y β . Al analizar de la misma forma el conjunto de redes cristalinas completo de un cierto compuesto, se genera la matriz de distancia, $\mathbf{D}$, que condensa la comparación por pares del conjunto de estructuras cristalinas moleculares (figs. 7 y 8). Esta matriz depende del número l de moléculas cercanas a la molécula de referencia, además de las métricas entre la posición y la orientación relativas de pares de moléculas (s , z , z_{rc} , *RMSD* y *RCD*). Por lo tanto, se calcularon matrices de distancia con diferentes valores de l y con diferentes métricas entre pares de moléculas, lo cual dependió de los objetivos de los análisis presentados en la sección de resultados. En consecuencia, las matrices de distancia empleadas se especificarán en dicha sección. El diagrama de flujo general del algoritmo para la comparación por pares de estructuras cristalinas de un mismo compuesto se muestra en la figura 8. Asimismo, las matrices de distancia calculadas para los conjuntos de estructuras de cada compuesto se emplearon para generar clústeres de estructuras cristalinas con métodos de agrupamiento, lo cual se discute en la siguiente sección.

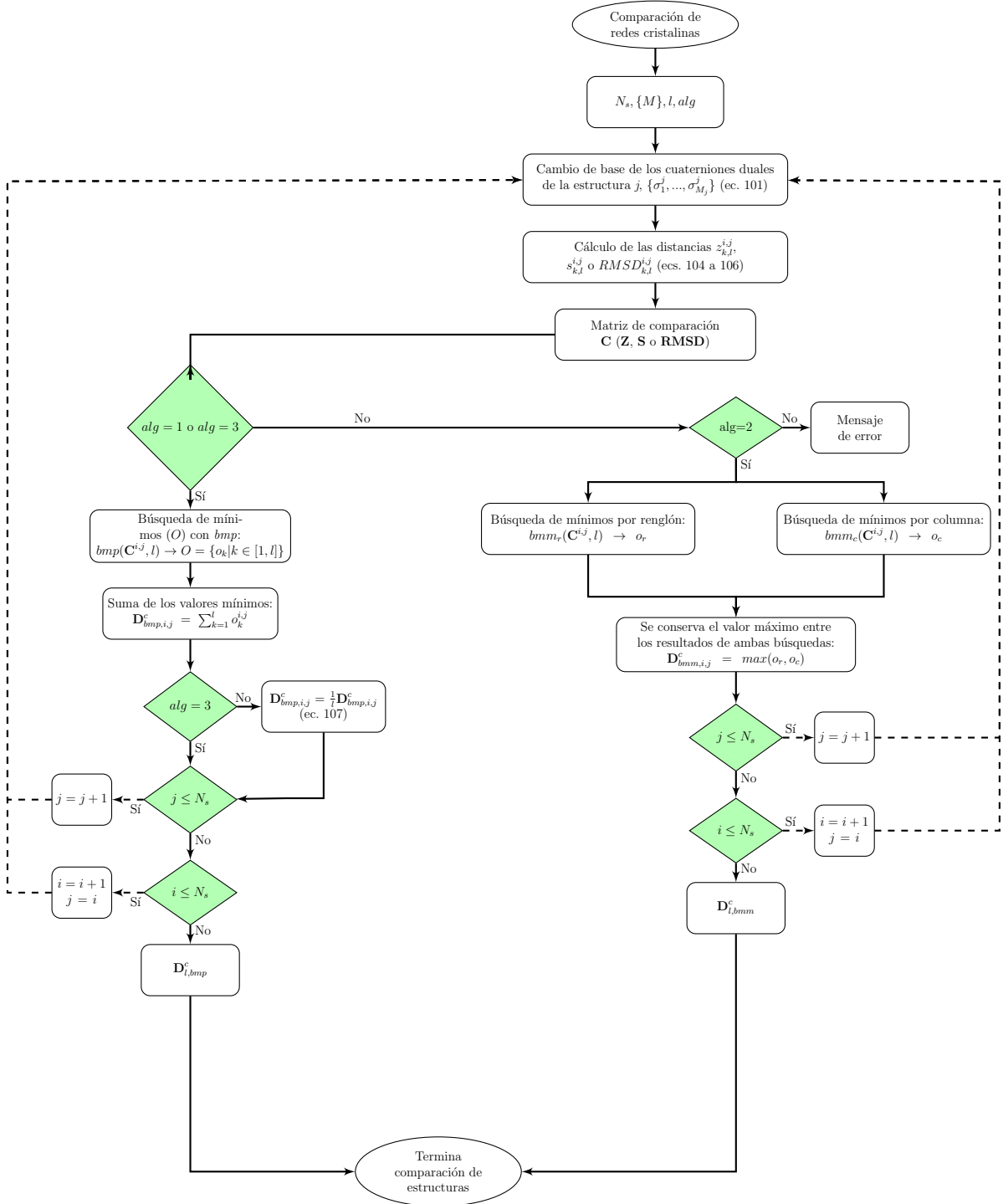


Figura 8: Diagrama de flujo mostrando la metodología empleada para la comparación de redes cristalinas moleculares generadas de forma aleatoria para los diferentes compuestos estudiados en esta tesis. Sea alg un parámetro que determina el tipo de algoritmo de búsqueda de mínimos a emplear: bmp , para el cual se consideran dos variaciones (1 y 3), y bmm (2).

En la generación de las matrices de distancia se consideraron en particular 8, 14, 26 y 107 moléculas vecinas (l) para cada métrica (s , z , RCD , $RMSD$). Esto se debe a que el primer límite es considerado como estándar en el cálculo de la matriz de distancia $\mathbf{D}^{RCD}$ (Li et al. 2018) y, por lo tanto, se mantuvo con el fin de comparar las demás métricas con esta última. En el caso del límite 107, este corresponde al número total de moléculas en la primera zona periódica ($z_p = 1$) para los sistemas con cuatro moléculas en la celda unitaria, como los de la formamida (ver tab. 1). Cabe mencionar que este valor de l genera las mejores agrupaciones con las métricas propuestas, lo cual se discute en los resultados. Por último, los límites 14 y 26 se seleccionaron considerando sistemas con dos y cuatro moléculas en la celda unitaria, pero tomando en cuenta solo los vecinos más cercanos en la primera zona periódica. Esto es, si consideramos un sistema ideal, tal que $|\vec{c}| < |\vec{a}|, |\vec{b}|$, y con dos moléculas en la celda unitaria como el que se muestra en la figura 9a, se generan en un plano 4 moléculas cercanas alrededor de cualquier molécula en la celda central, como la que se resalta en rojo. Extendiendo el análisis a tres dimensiones se tendrían 14 moléculas vecinas (5 moléculas más por encima del plano y 5 más por debajo). Al aplicar este mismo análisis a un sistema con cuatro moléculas en la celda unitaria (fig. 9b) se obtienen 26 moléculas: 8 en dos dimensiones, 9 arriba y 9 abajo. Así, estos dos últimos valores del número de moléculas vecinas se seleccionaron considerando que $|\vec{c}| < |\vec{a}|, |\vec{b}|$, con el fin de evitar la arbitrariedad en la elección de este parámetro (l). Además, como se verá en los resultados, esto fue necesario para poder entender mejor los clústeres generados a partir de las matrices de distancia, $\mathbf{D}_{l,bmp \text{ o } bmm}^C$.



Figura 9: Esquema mostrando los nodos o moléculas cercanas a la molécula de referencia en la celda central al replicar celdas unitarias con (a) $Z=2$ y (b) $Z=4$ moléculas, considerando que $|\vec{c}| < |\vec{a}|, |\vec{b}|$. Se resalta en rojo la molécula de referencia de la celda central y en líneas verdes discontinuas se muestran sus vecinos cercanos en dos dimensiones.

5.5. Agrupamiento

Para agrupar cada conjunto de estructuras cristalinas generado aleatoriamente para cada compuesto analizado en esta tesis, se utilizaron tres métodos de agrupamiento que dependen de las matrices de distancia descritas en la sección anterior: el espectral, de propagación de afinidades y de *OPTICS* (cf. sección 2.3). Estos métodos se implementaron utilizando la paquetería *scikit-learn* en *python* (Pedregosa et al. 2011), de acuerdo con los métodos descritos en la sección 2.3. En el caso de los dos primeros métodos de agrupamiento, se requieren emplear matrices de afinidad; por lo tanto, fue necesario transformar las matrices de distancia (ecs. 107 y 108) de acuerdo con la siguiente ecuación (von Luxburg 2007):

$$a_{ij} = \exp(-d_{ij}/\sigma) \quad (109)$$

donde a_{ij} son las entradas de la matriz de afinidad, d_{ij} son los valores asociados a la matriz de distancia y σ es un hiperparámetro que fue optimizado.

Posteriormente, en el caso del método espectral, a partir de la matriz de afinidad se calculó la matriz de grado y el laplaciano normalizado, de acuerdo con las ecuaciones 46 y 47 (cf. sección 2.3.2). A partir de lo anterior, se determinaron los eigenvalores más pequeños del laplaciano y se construyó una matriz con los eigenvectores correspondientes. Finalmente, empleando la matriz de eigenvectores se generaron los clústeres de las estructuras cristalinas en el conjunto de cada compuesto con el método estándar de *k-means* implementado en la librería *scikit-learn* (Pedregosa et al. 2011). Cabe mencionar que el número de eigenvectores considerados en la matriz de entrada fue igual al número de clústeres buscados con el método *k-means*, dada la relación establecida por los eigenvalores (cf. sección 2.3.2).

Dado que tanto el hiperparámetro σ como el número de clústeres no se conocen *a priori*, estos se modificaron hasta obtener un resultado óptimo y reproducible. El hiperparámetro σ se modificó entre 1 y 10^8 , con cambios de un orden de magnitud. Para cada valor de σ se aplicó el método de agrupamiento espectral, descrito en el párrafo anterior, y, además, se repitió para un número de clústeres entre 2 y 10. Para cada número de clústeres se calculó el puntaje de silueta promedio, empleando la paquetería *scikit-learn* (Pedregosa et al. 2011). Entre el conjunto de valores del coeficiente de silueta promedio se seleccionó el máximo y el número de clústeres correspondiente se identificó como óptimo. Tal que, se obtiene el número de clústeres óptimo con sus correspondientes puntajes de silueta para cada σ , además de otros datos asociados a cada clúster generado como los índices de las estructuras cristalinas que le son asignadas y la estructura cristalina más cercana al centro del clúster, la

cual será referida como *medoide*. Al final se obtiene un valor de silueta promedio, correspondiente al número de clústeres óptimo, para cada valor de σ . De este último conjunto de datos se conservan los resultados correspondientes a cualquier valor del coeficiente de silueta promedio que sea igual a la moda de los valores obtenidos para cada σ , lo cual asegura la reproducibilidad del agrupamiento al eliminar la arbitrariedad en la elección de este hiperparámetro. Esta parte de la metodología fue planteada con base en los resultados obtenidos con el método de agrupamiento espectral, lo cual se describe más adelante. Asimismo, esta metodología se resume en la figura 10, en la siguiente página.

Los métodos de propagación de afinidades y *OPTICS* se implementaron de manera directa de acuerdo con sus definiciones estándar en la paquetería de *scikit-learn* (Pedregosa et al. 2011). Para el primer método se utilizó la misma metodología para la transformación de las matrices de distancia que se usó para el método espectral: generación de las matrices de afinidad (ec. 109), y se varió el hiperparámetro *preferencia* hasta encontrar los clústeres óptimos; es decir, los clústeres que corresponden al mayor coeficiente de silueta promedio y, por lo tanto, generan la mejor separación de las estructuras cristalinas, según la matriz de distancia empleada. Para el segundo método se emplearon las matrices de distancia tal cual como fueron calculadas (ecs. 107 y 108) y se varió el hiperparámetro ξ (*cf.* sección 2.3.4). Asimismo, también para estos métodos de agrupamiento, se calculó el valor del coeficiente de silueta promedio (*cf.* sección 2.3.5), también contenido en la paquetería *scikit-learn*.

Cabe enfatizar que el término “clúster” en esta tesis hace referencia exclusivamente a un conjunto de estructuras cristalinas con configuraciones (arreglos de las moléculas en las redes cristalinas) o energías similares, según la matriz de distancia empleada con el método de agrupamiento. Así, de acuerdo con lo descrito en las secciones anteriores, la metodología desarrollada en esta tesis parte de generar un conjunto vasto de estructuras cristalinas aleatoriamente para un compuesto. Las estructuras cristalinas de este conjunto inicial son comparadas por pares a través del cálculo de una matriz de distancia, **D**. Finalmente, esta matriz es empleada para generar clústeres de estructuras cristalinas con configuraciones o energías similares con alguno de los tres métodos de agrupamiento mencionados en esta sección, el cual será especificado en los resultados. Esta metodología es aplicada con diferentes matrices de distancia y para cada conjunto de estructuras cristalinas generadas aleatoriamente para cada uno de los tres compuestos analizados en esta tesis (formamida, hidrazida maleica y ácido oxálico).

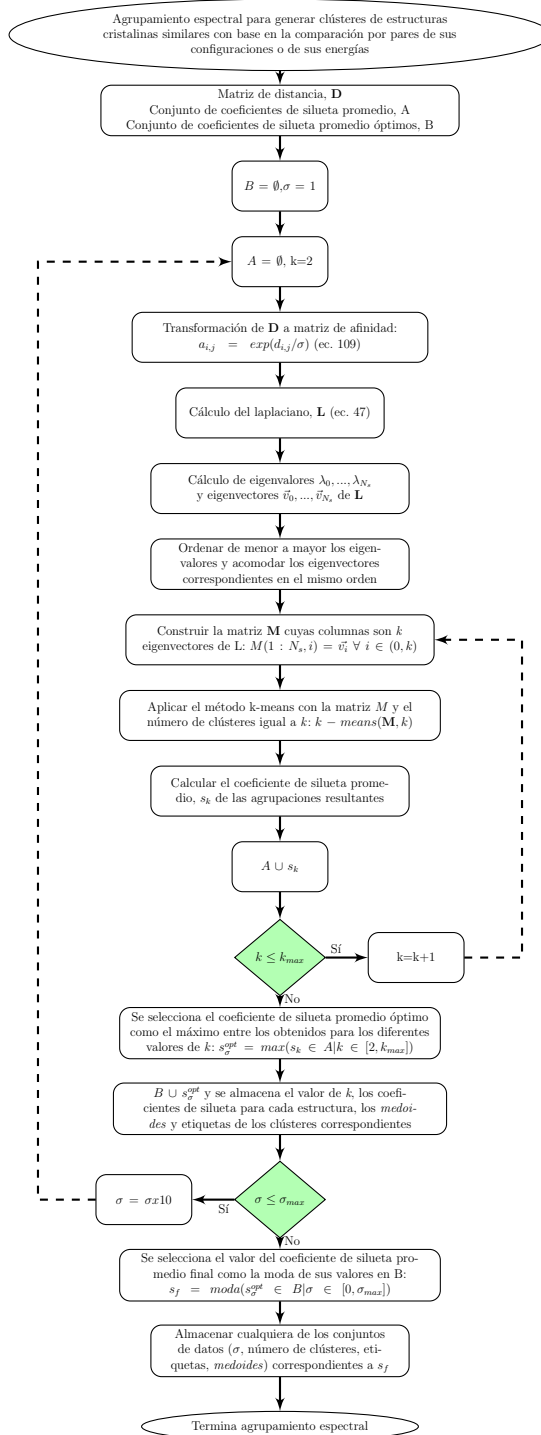


Figura 10: Diagrama de flujo del algoritmo desarrollado para el agrupamiento de un conjunto de estructuras cristalinas de un mismo compuesto molecular con base en la comparación por pares de sus configuraciones o de sus energías a través de matrices de distancia, **D**, basado en los métodos de agrupamiento espectrales.

5.6. Análisis de las métricas y de las agrupaciones

Entre los métodos de agrupamiento descritos en la sección anterior, solo el método espectral permitió obtener clústeres con coeficientes de silueta promedio altos, lo cual se presenta en la sección de resultados. Por lo tanto, en esta sección se describen análisis de los clústeres de estructuras cristalinas obtenidos solo con el método de agrupamiento espectral.

La metodología de agrupamiento espectral no identifica *medoides*; por lo tanto, fue necesario proponer una manera simple de obtenerlos. De acuerdo con los antecedentes (*cf.* sección 2.3), el método *k-means* encuentra los centroides de cada clúster (*cf.* sección 2.3.1), los cuales en el caso del método espectral son puntos en el espacio vectorial en el que se proyectan los datos (*spectral embedding*). Por ello, estos puntos no corresponden a ninguna estructura en particular, pero son representativos de los clústeres en el espacio vectorial generado. Por lo tanto, la estructura cristalina más cercana a un cierto centroide también es representativa del clúster correspondiente; es decir, esta estructura cristalina es el *medoide*. Así, en el espacio vectorial en el que se proyectan los datos, se calculan para un clúster las distancias euclidianas de todos los puntos (asociados a cada estructura cristalina) al centroide y la estructura asociada al punto con la menor distancia al centroide es identificada como el *medoide* del clúster analizado. Esto se logra al emplear la matriz de eigenvectores $\mathbf{M}$ descrita para el método espectral (*cf.* sección 2.3.2), ya que sus filas son los vectores en el espacio de proyección correspondientes a cada estructura cristalina. Así, al aplicar esta metodología para cada clúster generado con el método de agrupamiento espectral, se identifican los *medoides* para los clústeres óptimos de acuerdo con la metodología previamente descrita (fig. 10).

Cabe recordar que las matrices de distancia empleadas en el método de agrupamiento espectral contienen la distancia entre las configuraciones o las energías, según la matriz $\mathbf{D}$ calculada, de cada par de estructuras cristalinas en el conjunto analizado. Ahora, considerando que se conocen los *medoides* para cada clúster, es posible calcular el *RMSD* de las distancias entre los *medoides* y las demás estructuras cristalinas que conforman los clústeres correspondientes. Estas distancias se toman directamente de la matriz $\mathbf{D}$ empleada en el método de agrupamiento y no en el espacio vectorial generado con el *spectral embedding*, como en el caso descrito en el párrafo anterior. Así, de manera similar a la ecuación 88,

$$RMSD_k = \left(\frac{1}{n_k} \sum_{i \in G_k} d_{m_k, i}^2 \right)^{1/2} \quad (110)$$

donde G_k es el clúster considerado, n_k es el número de estructuras cristalinas contenidas en este y $d_{m_k,i}$ son los valores de la matriz de distancia empleada en el agrupamiento, tal que m_k es el índice del *medoide* correspondiente. Además, el *RMSD* de las distancias a los *medoides* se normaliza con el fin de eliminar el efecto del tamaño de los clústeres y de la escala de las diferentes métricas. De acuerdo con la literatura (Otto 2019), una de las mejores maneras de normalizar el *RMSD* es dividirlo entre el promedio de las distancias correspondientes, lo cual se conoce como coeficiente de variación:

$$CV(RMSD)_k = \frac{RMSD_k}{\bar{G}_k} \quad (111)$$

donde $\bar{G}_k$ es el promedio de las distancias entre las estructuras que conforman el clúster k y su *medoide* correspondiente, tal que

$$\bar{G}_k = \frac{1}{n_k} \sum_{i \in G_k} d_{m_k,i} \quad (112)$$

Resulta importante remarcar, que el $CV(RMSD)$ es una medida de la dispersión de las distancias entre las estructuras cristalinas de un clúster y su *medoide*; por lo tanto, no está relacionada con el *RMSD* empleado como una métrica para medir la distancia entre las configuraciones de pares de estructuras cristalinas (α y β , ec. 106). Así, la distinción entre estas dos maneras de emplear el *RMSD* se hará por el contexto del análisis o haciendo referencia explícita de las matrices $\mathbf{D}$ (ecs. 107 y 108) y del CV (ec. 111).

Por otra parte, se realizaron análisis de las energías electrónicas de las estructuras cristalinas que conforman cada clúster. Para cada clúster se calculó el promedio y la desviación estándar de las energías electrónicas de las estructuras cristalinas, y se identificó la energía más baja.

De manera adicional al análisis de los clústeres que resultan de aplicar el método de agrupamiento espectral, se calcularon matrices de distancia⁵ entre las estructuras experimentales y las estructuras cristalinas generadas de manera aleatoria en el caso de los sistemas que presentan polimorfismo (*MH* y *OA*). Así, se identificaron las estructuras cristalinas con configuraciones moleculares más similares a las experimentales (son aquellas con las que se obtuvieron las menores distancias en las matrices $\mathbf{D}$). Finalmente, se calcularon las diferencias entre los vectores que describen la celda unitaria de las estructuras cristalinas experimentales y aquellas que fueron identificadas como configuracionalmente semejantes. Para esto se promedia-

⁵Las matrices de distancia empleadas se especifican en los resultados.

ron las distancias euclidianas entre los vectores de la celda unitaria equivalentes:

$$\Delta\alpha = \frac{1}{3} \sum_{i=1}^3 |\vec{\alpha}_i^{exp} - \vec{\alpha}_i| \quad (113)$$

donde $\vec{\alpha}_i^{exp}$ es el vector de la celda unitaria en la dirección i de la estructura experimental y $\vec{\alpha}$ el vector correspondiente en la estructura cristalina más semejante dentro del conjunto de estructuras generado aleatoriamente. Asimismo, se calculó la desviación relativa en porcentaje entre los volúmenes de la celda unitaria de la estructura cristalina más semejante, V , respecto al de la experimental, V^{exp} :

$$\Delta V = 100 \frac{V - V^{exp}}{V^{exp}} \quad (114)$$

6. Resultados

De acuerdo con lo descrito en la metodología, en este trabajo se desarrollaron diferentes programas para describir y comparar la configuración de moléculas en redes cristalinas, que se generaron aleatoriamente, empleando cuaterniones duales. Se analizaron 15594, 1092 y 792 estructuras cristalinas generadas con el programa *Genarris* para la formamida, el ácido oxálico y la hidrazida maleica, respectivamente. En la primera parte de esta sección se presenta un análisis de los algoritmos propuestos para la representación y comparación de la configuración (arreglo espacial de las moléculas) de redes cristalinas empleando cuaterniones duales y se verifica que estos funcionen de manera adecuada. Asimismo, se examina la posibilidad de generar clústeres de estructuras cristalinas semejantes con diferentes métodos de agrupamiento a partir de algunas de las matrices de distancia propuestas en la metodología, con el fin de identificar el método de agrupamiento adecuado para el tipo de datos analizados. Posteriormente, cada una de las métricas propuestas en este trabajo (z , z_{rc} y s , ecs. 104 y 105) para medir la distancia entre pares de cuaterniones duales, que representan la orientación relativa de pares de moléculas, es comparada con el *RMSD* de las distancias entre las coordenadas de los átomos de pares de moléculas (ec. 106) con el fin de seleccionar la métrica que genere la mejor descripción del ambiente químico alrededor de una molécula de referencia. Finalmente, se estudian los clústeres de estructuras cristalinas con configuraciones similares generados con el método de agrupamiento espectral a partir de diferentes matrices de distancia calculadas con la métrica z ($\mathbf{D}^z$), identificada como la mejor métrica en comparación con z_{rc} y s . Los clústeres obtenidos con las matrices $\mathbf{D}^z$ se comparan con los clústeres que resultan de emplear las matrices de distancia calculadas con las métricas *RCD* y *RMSD*. Además, se comparan las estructuras cristalinas determinadas experimentalmente de la hidrazida maleica y del ácido oxálico con las estructuras de los conjuntos generados aleatoriamente que fueron identificadas como semejantes con diferentes matrices de distancia, con el fin de corroborar la validez de las métricas para identificar estructuras con configuraciones similares.

Cabe mencionar que en los análisis presentados para las estructuras cristalinas de la formamida se emplearon subconjuntos de estas estructuras. Estas estructuras forman parte del conjunto de 15594 estructuras cristalinas generadas de manera aleatoria para la formamida. Pero los subconjuntos no representan estructuras cristalinas con configuraciones o energías similares, a diferencia de los clústeres que sí representan un análisis de la similitud (según la matriz de distancia empleada) entre las estructuras con el método de agrupamiento espectral. Por ejemplo, un subcon-

junto de 3000 estructuras cristalinas de la formamida implica que se tomaron las primeras 3000 estructuras en el conjunto completo de 15594 estructuras, las cuales posteriormente fueron analizadas con diferentes matrices de distancia y el método de agrupamiento espectral. El análisis de los subconjuntos se realizó debido a que no siempre fue computacionalmente factible analizar el conjunto completo de estructuras cristalinas (15594). Asimismo, se evaluó el efecto del tamaño del subconjunto analizado sobre los resultados del método de agrupamiento espectral. Sin embargo, como se presenta más adelante, al final se realizó un análisis del conjunto completo de las estructuras cristalinas de la formamida; además, no se encontraron discrepancias entre los resultados obtenidos con los subconjuntos y los obtenidos con el conjunto completo. Esto solo concierne a las estructuras de la formamida, ya que en los casos de la hidrazida maleica y del ácido oxálico se generaron conjuntos de estructuras con *Genarris* de menor tamaño y, por lo tanto, no fue necesario analizar subconjuntos.

6.1. Verificación de los algoritmos

Previo a la validación de las métricas para la comparación de la configuración de moléculas en estructuras cristalinas se verificó el correcto funcionamiento de los programas empleados en el cálculo de estas. Para ello se aplicaron los algoritmos para representar y comparar la configuración de estructuras cristalinas con cuaterniones duales (figs. 6 y 8) y para el cálculo de matrices de distancia (fig. 7) a una estructura cristalina de la formamida y la misma estructura rotada. Esto con el fin de corroborar numéricamente que las métricas fuesen invariantes ante la rotación de los sistemas. Así, como ejemplo, se consideró una rotación de 90° alrededor del eje x o $\vec{a}_1$ en coordenadas cartesianas, tal que la imagen rotada se obtuvo simplemente intercambiando las coordenadas y ($\vec{a}_2$) por $-z$ ($-\vec{a}_3$) y z ($\vec{a}_3$) por y ($\vec{a}_2$) de la estructura cristalina original. Al aplicar las métricas ($\mathbf{D}^z$, $\mathbf{D}^s$, $\mathbf{D}_{bmp}^{RCD}$, $\mathbf{D}^{RMSD}$) sobre este par de estructuras cristalinas de prueba se obtuvieron matrices nulas en todos los casos, considerando los algoritmos *bmp* y *bmm* en cada caso. Así, las medidas cumplen con ser invariantes ante la rotación y se tiene un indicio de que los algoritmos funcionan correctamente. Además, esto es una prueba numérica de que el cambio de base de cuaterniones duales planteado en la metodología (ec. 101) es correcto.

Por otra parte, se verificó el funcionamiento correcto del algoritmo para representar la posición y orientación en relación a una molécula de referencia de moléculas en redes cristalinas con cuaterniones duales (fig. 6). De acuerdo con lo descrito en la metodología, en una estructura cristalina cada cuaternión dual (σ_i) representa

Compuesto	RMSD promedio / Å	RMSD máximo / Å	N_s
Formamida	0.0019	0.0059	15594
Ácido oxálico	0.0075	0.04	970
Hidrazida maleica	0.042	0.13	1092

Tabla 3: Valores promedio y máximo del *RMSD* de las distancias entre las coordenadas de todos los átomos de las moléculas obtenidas al generar el ambiente químico alrededor de la celda unitaria y las que resultan de aplicar los cuaterniones duales correspondientes sobre la molécula de referencia en cada estructura cristalina generada de manera aleatoria para cada compuesto analizado en esta tesis. El valor promedio del *RMSD* se calculó primero sobre las M moléculas en la primera zona periódica de cada estructura y los valores se promediaron a su vez sobre todas las redes cristalinas (N_s). El valor máximo se obtuvo considerando todas las transformaciones y todas las estructuras cristalinas en el conjunto generado aleatoriamente.

la transformación (rotación más traslación) de la molécula de referencia tal que las coordenadas de sus átomos coincidan con las coordenadas de los átomos de la molécula que el cuaternión dual representa: $v_a^i = \sigma_i \otimes v_a^{ref} \otimes \sigma_i^\diamond$, donde v_a^i son las coordenadas del átomo a en la molécula i y v_a^{ref} son las coordenadas del átomo equivalente en la molécula de referencia, ambas expresadas como cuaterniones duales puros ($v = (1, 0, 0, 0) + \epsilon(0, \vec{v})$). Tal que si el algoritmo para calcular los cuaterniones duales funciona correctamente, no deben existir diferencias entre las coordenadas de los átomos leídas del archivo de entrada (incluyendo las obtenidas al generar las traslaciones periódicas de la celda unitaria) y las obtenidas al aplicar los cuaterniones duales correspondientes sobre la molécula de referencia, dentro de un cierto intervalo de error numérico. De acuerdo con la metodología, al aplicar en una estructura cristalina cada cuaternión dual sobre la molécula de referencia se generan en la primera zona periódica ($z_p = 1$) M moléculas ($M = 107$ para $Z = 4$ y $M = 53$ para $Z = 2$); para cada una de estas moléculas se calcula el *RMSD* de las distancias entre las coordenadas de todos sus átomos obtenidas al generar el ambiente químico alrededor de la celda central y las coordenadas de los átomos equivalentes que resultan de aplicar el cuaternión dual correspondiente sobre la molécula de referencia. Para cada estructura cristalina en el conjunto generado para un compuesto se promedió el *RMSD* de las distancias antes mencionadas para las M moléculas y los valores de todo el conjunto de estructuras a su vez fueron promediados (tab. 3). También se determinó el *RMSD* máximo en el conjunto de datos obtenido al aplicar todo el conjunto de cuaterniones duales (tab. 3).

Los valores del *RMSD* de las distancias entre las coordenadas calculadas originalmente al trasladar periódicamente la celda unitaria (generación de ambiente químico) y las coordenadas generadas al aplicar los cuaterniones duales correspon-

dientes sobre la molécula de referencia (tab. 3) son menores o similares al error promedio estimado para las coordenadas atómicas medidas experimentalmente por difracción de rayos X de monocristales de cristales moleculares: alrededor de 0.1 Å (Brunger 1997, van de Streek y Neumann 2010). Solo en el caso de la hidrazida maleica (*MH*) el *RMSD* máximo llega a ser ligeramente superior al error experimental de las coordenadas atómicas (0.1 Å). Esto se debe a que, entre los compuestos analizados, la hidrazida maleica es la molécula conformada por el mayor número de átomos. Así, esta mayor diferencia puede atribuirse a una propagación del error numérico en el cálculo del *RMSD*; sin embargo, esto es un caso particular, ya que el *RMSD* promedio para las estructuras de la hidrazida maleica es mucho menor (tab. 3). En consecuencia, el error cometido con los cuaterniones duales, en las estructuras cristalinas analizadas, es despreciable; aunque no se descarta que en otras estructuras este pueda llegar a ser mayor, en cuyo caso sería necesario optimizar las transformaciones propuestas, de acuerdo con lo descrito en el marco teórico (*cf.* sección 2.4). Aun así, los valores del *RMSD* obtenidos en las estructuras cristalinas consideradas en este trabajo (tab. 3) muestran que el algoritmo para la generación de cuaterniones duales funciona de manera adecuada.

En el caso del algoritmo *bmm* existe una indeterminación al momento de ordenar los cuaterniones duales de cada estructura. De acuerdo con la metodología *bmm* (fig. 7b), el conjunto de cuaterniones duales generado para cada estructura cristalina es reducido a los l cuaterniones duales con las menores distancias $|\vec{T}|$ en relación a la molécula de referencia. Al reducir el conjunto de cuaterniones duales, puede que sea necesario elegir entre dos cuaterniones duales que correspondan a la misma distancia $|\vec{T}|$, tal que cambios en la precisión del cálculo pueden resultar en una variación del conjunto de cuaterniones duales final para algunas estructuras cristalinas. Esto se observó al emplear dos compiladores diferentes para el algoritmo *bmm*: *ifortran* y *gfortran*. Es decir, el mismo código compilado de dos maneras diferentes generó conjuntos de cuaterniones duales distintos, debido a que la precisión numérica es distinta para cada compilador. Por lo tanto, se evaluó el efecto del compilador al generar las matrices de distancia de la siguiente forma; se calculó la matriz $\mathbf{D}_{bmm}^z$ dos veces, una vez empleando cada compilador, para 14 subconjuntos de estructuras cristalinas de la formamida variando el número de estructuras entre 2000 y 15594 y el número de moléculas cercanas a la molécula de referencia ($l = 14, 26, 32$ y 107). Posteriormente, cada matriz fue empleada para generar clústeres de las estructuras cristalinas del subconjunto analizado con el método de agrupamiento espectral. Al emplear el método de agrupamiento espectral con las matrices de distancia $\mathbf{D}_{bmm}^z$ se obtiene para cada una de estas un conjunto de números enteros, denominados

etiquetas, que definen el clúster al cual fue “asignada” cada estructura. En todos los casos, el número de clústeres óptimo fue de dos para las matrices correspondientes a ambos compiladores; es decir, los valores de las etiquetas (números enteros) asignadas fueron cero o uno. De este modo, la diferencia entre los clústeres obtenidos empleando el mismo tipo de matriz $\mathbf{D}_{bmm}^z$, pero calculada de dos maneras distintas, fue simplemente el promedio del cuadrado de las diferencias entre las etiquetas obtenidas. En otras palabras, se determinó el promedio del número de estructuras que fueron “asignadas” a clústeres diferentes. En la figura 11 se puede observar que en general las diferencias entre las etiquetas obtenidas para cada matriz $\mathbf{D}_{bmm}^z$ son menores al 1 %. En particular, en el caso de la matriz $\mathbf{D}_{26,mm_s}^z$ para un subconjunto de 5000 estructuras cristalinas se obtuvo una diferencia mayor al 5 %. Sin embargo, esto fue un caso único con un valor aún relativamente bajo. Por consiguiente, las diferencias debidas a la ambigüedad en la elección de los cuaterniones duales con menor distancia $|\vec{T}|$ al momento de reducir el conjunto de cuaterniones duales para el algoritmo *bmm* son mínimas y el resto de los análisis se realizaron considerando solo el compilador *gfortran*.

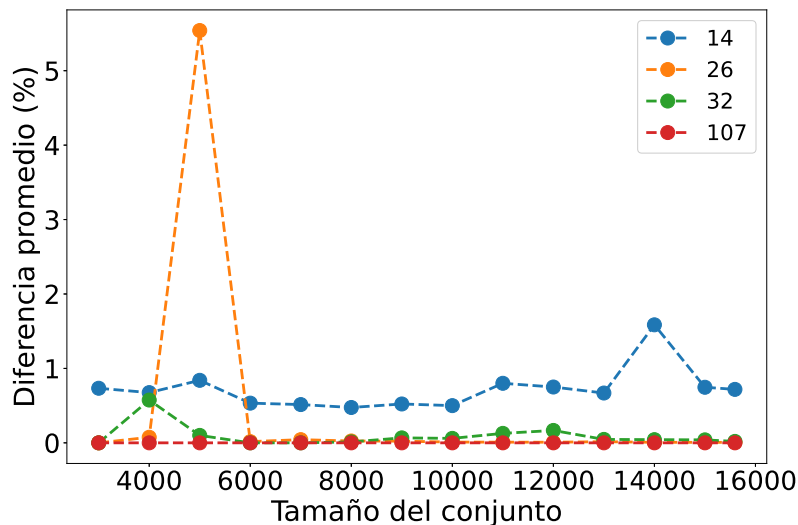


Figura 11: Diferencia en porcentaje de las etiquetas asignadas a las estructuras cristalinas de la formamida agrupadas con el método de agrupamiento espectral a partir de las matrices $\mathbf{D}_{bmm}^z$ calculadas con el algoritmo *bmm* compilado con *gfortran* y con *ifortran* para conjuntos de estructuras cristalinas de la formamida de diferente tamaño. Las matrices se calcularon por cada compilador con el número de moléculas cercanas a la molécula de referencia igual a 14 (azul), 26 (naranja), 32 (verde) y 107 (rojo).

Además, se analizó el efecto de escalar los datos con el factor $1/l$ en el cálculo de las matrices de distancia con el algoritmo *bmp* (ec. 107), ya que en el cálculo de $\mathbf{D}^{RCD}$ este factor no es considerado (Li et al. (2018), ec. 91). Para todos los métodos de agrupamiento empleados no se observó ninguna diferencia en los resultados al

incluir u omitir este factor. Así, dado que al considerar el factor $1/l$ se genera un escalamiento homogéneo de los datos de la matriz $\mathbf{D}^{RCD}$, este no afecta los clústeres generados. Además, en el caso de los métodos de agrupamiento que emplean matrices de afinidad, la transformación de las matrices de distancia empleada (ec. 109) es también una normalización de los datos; por lo tanto, cualquier escalamiento previo de la matriz de distancia se pierde en la matriz de afinidad. En consecuencia, los resultados obtenidos con las matrices $\mathbf{D}^{RCD}$, que fueron calculadas sin el factor $1/l$ (ec. 91), son completamente equivalentes al caso de haber incluido este factor en el cálculo de las mismas (ec. 107). Por lo tanto, esta medida será referida como $\mathbf{D}_{bmp}^{RCD}$, aunque no se incluya el factor $1/l$ propiamente en el cálculo, como se muestra en la ecuación 107. Esto último con el fin de mantener la definición propuesta por Li et al. (2018) y, adicionalmente, para enfatizar su relación con las demás matrices de distancia que se generen empleando el algoritmo *bmp*.

Así, los análisis presentados en esta sección permitieron descartar errores de los programas desarrollados para describir con conjuntos de cuaterniones duales la configuración de moléculas en redes cristalinas y para el cálculo de matrices de distancia. Por lo tanto, los cuaterniones duales son una herramienta matemática que permite describir correctamente la configuración de moléculas en redes cristalinas.

6.2. Análisis de métodos de agrupamiento

Dado que la eficacia de los algoritmos de agrupamiento depende del tipo de datos analizados, fue necesario emplear diferentes métodos con el fin de identificar el más adecuado para la agrupación de las estructuras cristalinas analizadas en esta tesis. Para este análisis se consideraron solamente las matrices de distancia $\mathbf{D}_{107,bmp}^z$ y $\mathbf{D}_{8,bmp}^{RCD}$, ya que el objetivo fue únicamente evaluar la posibilidad de obtener clústeres de estructuras cristalinas con cada método de agrupamiento, independientemente de la matriz de distancia empleada. Asimismo, se consideró solo el subconjunto de las primeras 3000 estructuras del conjunto completo de estructuras cristalinas de la formamida (15594 estructuras). Este subconjunto se eligió con el fin de tener una muestra suficientemente representativa de las estructuras a analizar sin tener que requerir de un tiempo de cómputo muy prolongado para el cálculo de las matrices de distancia y para la aplicación de los métodos de agrupamiento. Además, este subconjunto fue suficiente para la validación de los métodos de agrupamiento, como se presenta a continuación.

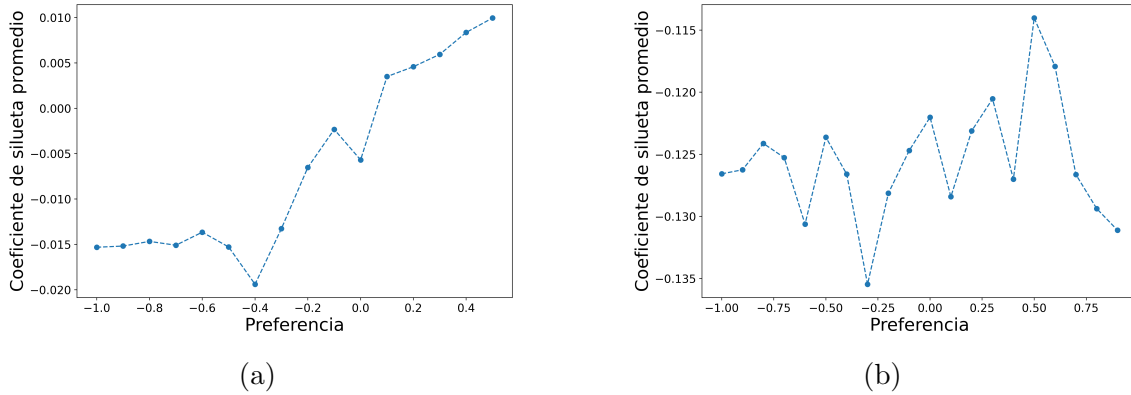


Figura 12: Coeficiente de silueta promedio de los clústeres obtenidos con el método de propagación de afinidades utilizando diferentes valores del hiperparámetro de *preferencia* empleando las matrices (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ de un subconjunto de 3000 estructuras cristalinas de la formamida generadas aleatoriamente.

Por una parte, para cada matriz de distancia ($\mathbf{D}_{107,bmp}^z$ y $\mathbf{D}_{8,bmp}^{RCD}$) se empleó el método de propagación de afinidades (*cf.* sección 2.3.3). Para ello, se transformaron las matrices de distancia en matrices de afinidad, de acuerdo con lo descrito en la metodología (ec. 109). Además, se varió el hiperparámetro de *preferencia* (valor en la diagonal de la matriz de afinidad) entre -1.0 y 2.0, con incrementos de 0.1. De acuerdo con la figura 12, el método de propagación de afinidades no es adecuado para el tipo de datos analizados. Sin importar los valores del hiperparámetro de *preferencia*, se obtienen valores del coeficiente de silueta negativos o muy cercanos a cero. Además, para valores de *preferencia* por encima o por debajo de los reportados

en las gráficas de la figura 12, el método no converge o es difícil encontrar los valores del hiperparámetro que lleven a la convergencia. Cabe mencionar que de acuerdo con el artículo de Li et al. (2018) el método de propagación de afinidades genera buenos resultados empleando directamente la matriz de distancia $\mathbf{D}_{8,bmp}^{RCD}$; sin embargo, el método de propagación de afinidades está ideado para emplearse con matrices de afinidad (Frey y Dueck 2007). Así, no se encontró una justificación para el uso de las matrices de distancia de forma directa. Por lo tanto, se descartó el método de propagación de afinidades para la agrupación de estructuras cristalinas.

El método *OPTICS* (cf. sección 2.3.4) se aplicó empleando de forma directa las matrices de distancia previamente mencionadas. En este caso se varió el hiperparámetro ξ , el cual delimita los datos que pueden asociarse a un mismo clúster con base en sus valores de *alcanzabilidad* (cf. sección 2.3.4). Sin embargo, no fue posible generar agrupaciones con las matrices empleadas, ya que simplemente el método no convergió en ningún caso. Esto puede deberse a que es necesario realizar un tratamiento adicional de los datos como, por ejemplo, normalizar las matrices de distancia; sin embargo, esto no se exploró debido a que se obtuvieron mejores resultados con el método de agrupamiento espectral y este último fue suficiente para cumplir con los objetivos de esta tesis, como se discutirá más adelante. Así, el método de agrupamiento *OPTICS* también fue descartado.

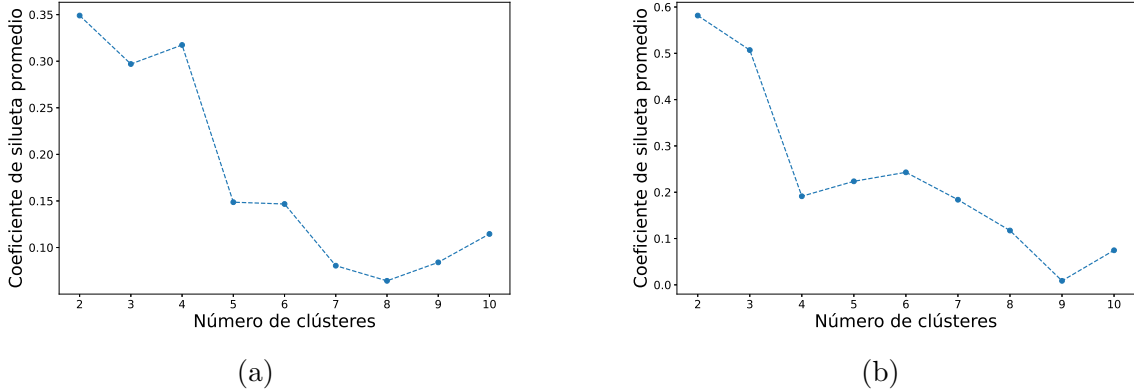


Figura 13: Coeficientes de silueta promedio obtenidos en la variación del número de clústeres al emplear las matrices de distancia (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ y, como método de agrupamiento, el método espectral seguido de *k-means* de un subconjunto de 3000 estructuras cristalinas de la formamida.

Finalmente, se evaluó el método espectral variando el parámetro σ en la transformación de las matrices de distancia a matrices de afinidad (ec. 109) y variando el número de clústeres. Esto con el fin de determinar los valores de σ y de número de clústeres que generen los mejores resultados (valores más altos del coeficiente de silueta promedio). El parámetro σ se varió entre 10^{-17} y 10^{17} con incrementos de una unidad en el exponente; a su vez, para cada valor de σ se varió el número de

clústeres entre 2 y 10, de acuerdo con la metodología descrita en la figura 10. En el caso de variar el número de clústeres para un dado σ , se obtuvieron los resultados mostrados en la figura 13. De acuerdo con los antecedentes (*cf.* sección 2.3.5, Starczewski y Krzyżak (2015)), el valor máximo del coeficiente de silueta corresponde al número de clústeres óptimo para el conjunto de estructuras cristalinas analizado, este dato es justamente lo que se pudo obtener a partir de este tipo de análisis. Por ejemplo, para la matriz $\mathbf{D}_{107,bmp}^z$ (fig. 13a) el valor máximo del coeficiente de silueta promedio es de 0.35, lo cual corresponde a un número óptimo de clústeres de dos. Así, considerando lo anterior, se planteó parte del algoritmo descrito en la figura 10. Es decir, se conservan únicamente el número de clústeres, las etiquetas y los *medoides* asociados al valor máximo del puntaje de silueta promedio para cada valor de σ .

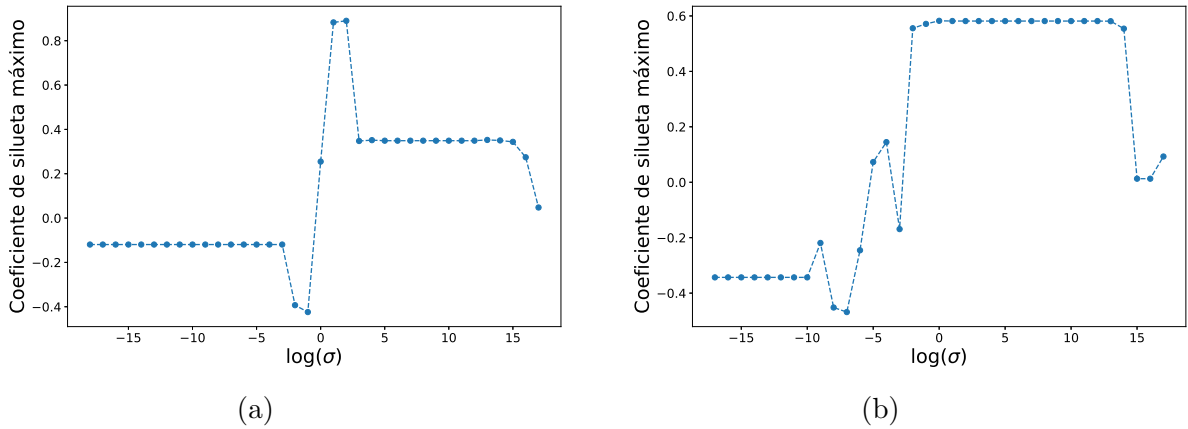


Figura 14: Coeficiente de silueta promedio máximo (respecto a los valores obtenidos para diferentes números de clústeres) para diferentes valores de σ al emplear las matrices de distancia (a) $\mathbf{D}_{107,bmp}^z$ y (b) $\mathbf{D}_{8,bmp}^{RCD}$ y el método espectral seguido de *k-means* para un subconjunto de 3000 estructuras cristalinas de la formamida.

De acuerdo con el párrafo anterior, se reporta en la figura 14 el valor máximo del coeficiente de silueta promedio para diferentes valores de σ . En estas gráficas (fig. 14) se puede observar que para ambas matrices de distancia ($\mathbf{D}_{107,bmp}^z$ y $\mathbf{D}_{8,bmp}^{RCD}$) existen rangos de valores de σ para los cuales el puntaje de silueta máximo se mantiene constante; es decir, el resultado del agrupamiento es el mismo. Así, es posible eliminar la dependencia del método espectral de este hiperparámetro al seleccionar cualquiera de los valores de σ que correspondan al mismo resultado. Sin embargo, también es importante considerar que para $\log(\sigma) < 0$ se obtienen valores negativos del coeficiente de silueta promedio; esto se debe a que, debido a la función exponencial empleada al transformar las matrices de distancia (ec. 109), los valores de afinidad se aproximan a cero al disminuir σ ($\log(\sigma) < 0$); es decir, se pierde la información de las relaciones entre las estructuras cristalinas, lo cual dificulta generar las

agrupaciones. Por lo tanto, para los siguientes análisis los cambios del hiperparámetro σ se mantendrán mayores o iguales a 1 ($\log(\sigma) \geq 0$), como se describió en la metodología. Además, cabe mencionar que para $\mathbf{D}_{107,bmp}^z$ existen valores del puntaje de silueta promedio por encima de aquellos que se mantienen constantes (fig. 14a). Sin embargo, estos valores no representan agrupaciones reales, ya que en estos casos se obtuvo un clúster con todas las estructuras cristalinas, excepto una que formó de manera “artificial” otro clúster, lo cual generó valores tan altos del coeficiente de silueta promedio erróneamente (*cf.* sección 2.3.5). Así, considerando lo anterior y dada la convergencia observada en el coeficiente de silueta promedio (fig. 14), se propone la metodología descrita en la figura 10 para el resto de los análisis.

6.3. Análisis de las métricas para medir la disimilitud entre configuraciones cristalinas

Con el fin de validar las métricas z , z_{rc} , s y RCD , se estudió su correlación con la medida $RMSD$, la cual es ampliamente usada en la comparación de sistemas químicos (cf. sección 2.4.2). Para esto, se analizó una única estructura cristalina de la formamida generada de manera aleatoria, en particular el conjunto de 107 moléculas alrededor de la molécula de referencia, considerando la primera zona periódica alrededor de la celda unitaria central (cf. Metodología). La estructura cristalina de la formamida fue elegida de manera aleatoria entre el conjunto de 15594 estructuras generadas con *Genarris*, ya que en esta sección el objetivo es analizar únicamente las métricas entre moléculas y no la comparación de la configuración de las estructuras cristalinas. Se calcularon las métricas (z , s , RCD y $RMSD$) entre la primera molécula del conjunto de 107 moléculas (cf. sección 5.3.1) y el resto. De acuerdo con la metodología presentada en la figura 8, lo anterior implica comparar la estructura cristalina considerada consigo misma, generar la matriz de comparación $\mathbf{C}$ (comparación de todos los pares posibles entre las 107 moléculas) y analizar el primer renglón de la matriz $\mathbf{C}$ (comparación de la primera molécula con el resto), para cada métrica. Así, se genera la comparación de un mismo conjunto de moléculas con diferentes métricas o medidas de disimilitud.

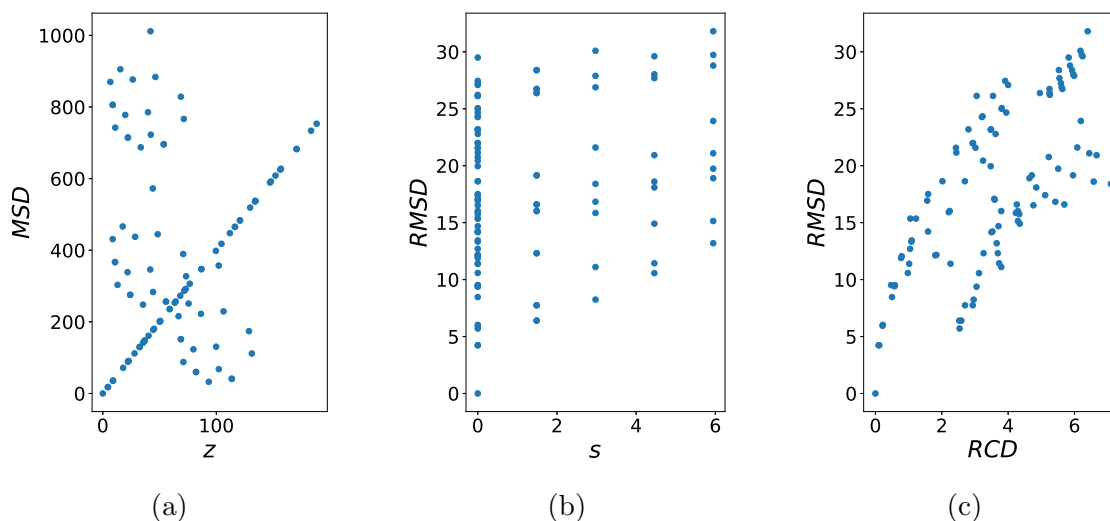


Figura 15: $RMSD$ de 106 moléculas de una de las estructuras cristalinas de la formamida, generadas aleatoriamente, respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en relación con las métricas (a) z , (b) s y (c) RCD . En el caso de la gráfica (a) se muestra el valor cuadrado del $RMSD$, sea el MSD , con el fin de mostrar de manera más clara la relación entre esta medida y z .

En la figura 15a se puede observar que existe una relación lineal entre el cuadrado

del *RMSD* (*MSD*) y z para una cantidad considerable de puntos (36 de 106). Así, la medida z permite generar una descripción de las posiciones y orientaciones relativas de las moléculas de manera similar al *MSD*. Evidentemente, si se toma la medida z_{rc} , que es la raíz cuadrada de z , esta muestra una relación lineal con el *RMSD*; aun así, se mantuvo el análisis con la medida z debido a que esta generó mejores resultados, como se verá más adelante. Sin embargo, de igual manera existen diferencias considerables entre z y el *MSD*. La mayoría de los puntos que se alejan de la línea central se encuentran por encima de esta (fig. 15a), lo cual implica que la métrica z detecta una mayor similitud para esas moléculas que el *MSD*.

Por el contrario, en el caso de la medida s , se puede observar en la figura 15b que no existe una relación directa entre esta y el *RMSD*. En general, la medida s asigna solo cinco valores distintos a las 106 moléculas, mientras que el *RMSD* muestra una mayor diversidad en las configuraciones de las mismas. Así, se tiene una primera evidencia de que la medida s falla como descriptor del espacio configuracional (espacio matemático que describe los diferentes arreglos espaciales de las moléculas en una red cristalina), tal que z es la mejor métrica entre las propuestas. Además, cabe mencionar que el hecho de que la medida s sea cero para la mayoría de las moléculas (fig. 15b) se debe a que esta no detecta diferencias entre moléculas con la misma orientación, aunque tengan diferente posición; la demostración de esto último se encuentra en el Anexo B (pág. 170). Esto implica que si el algoritmo para construir las matrices de distancia no captura los valores diferentes de cero para esta métrica, entonces, las matrices generadas tampoco permitirán comparar estructuras cristalinas, como se verá más adelante.

Por último, la métrica *RCD* no muestra una relación lineal con el *RMSD* tan evidente como la medida z (fig. 15c); sin embargo, esta medida muestra también una diversidad de valores para las diferentes moléculas y sigue manteniendo cierta correlación con el *RMSD*. Por lo tanto, las medidas z y *RCD* muestran comportamientos similares al *RMSD*; sin embargo, existen diferencias importantes que serán determinantes al momento de agrupar las estructuras cristalinas. Este mismo comportamiento se observó para una estructura del ácido oxálico y una de la hidrazida maleica (Anexo A, pág. 160). Aun así, esto es un indicio de que las métricas z y *RCD* se acercan a una descripción adecuada del espacio configuracional de moléculas en redes cristalinas.

Por otra parte, se analizó el efecto de los algoritmos *bmp* y *bmm* al calcular las matrices de distancia **D** para la comparación de pares de estructuras cristalinas. En este caso, se emplearon datos que condensan la comparación entre las configuraciones moleculares de pares de redes cristalinas y no se analiza la comparación por pares

de moléculas, como en las gráficas anteriores. Para cada matriz de distancia ($\mathbf{D}^z$, $\mathbf{D}^s$, $\mathbf{D}_{bmp}^{RCD}$, $\mathbf{D}^{RMSD}$) se consideró un subconjunto de 200 estructuras cristalinas de la formamida y se reportan los valores obtenidos para la comparación entre la primera estructura en el conjunto con las estructuras restantes (valores en el primer renglón de las matrices de distancia). Se seleccionó un subconjunto pequeño de estructuras cristalinas con el fin de generar gráficas que no contuviesen tantos puntos y fuese más fácil analizarlas visualmente. Cabe mencionar que en el caso de z la comparación se hizo entre los datos de las matrices $\mathbf{D}^z$ y $\mathbf{D}^{MSD}$ (figs. 16 y 17), no con $\mathbf{D}^{RMSD}$ como para las otras métricas; esto de acuerdo con la relación encontrada en el análisis anterior (fig. 15).

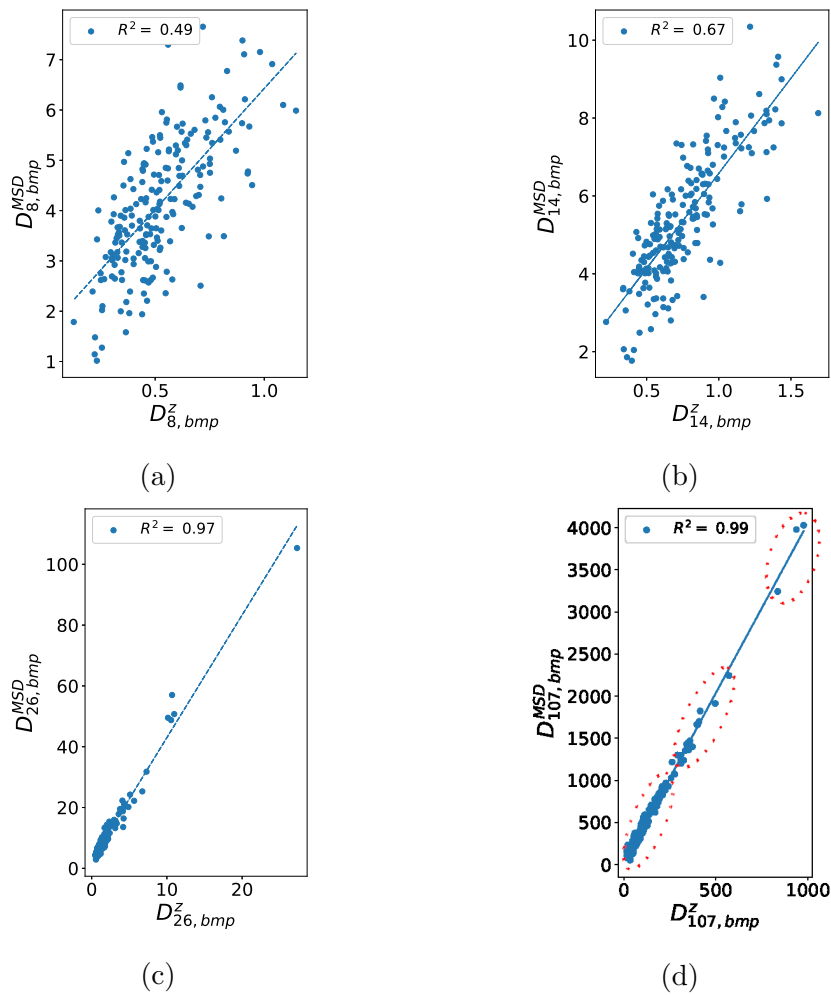


Figura 16: Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^z$ que fueron obtenidos con el algoritmo *bmp* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. En la figura (d) se muestran con elipses discontinuas en color rojo tres regiones separadas de puntos de manera aproximada que podrían representar clústeres de las estructuras cristalinas de la formamida.

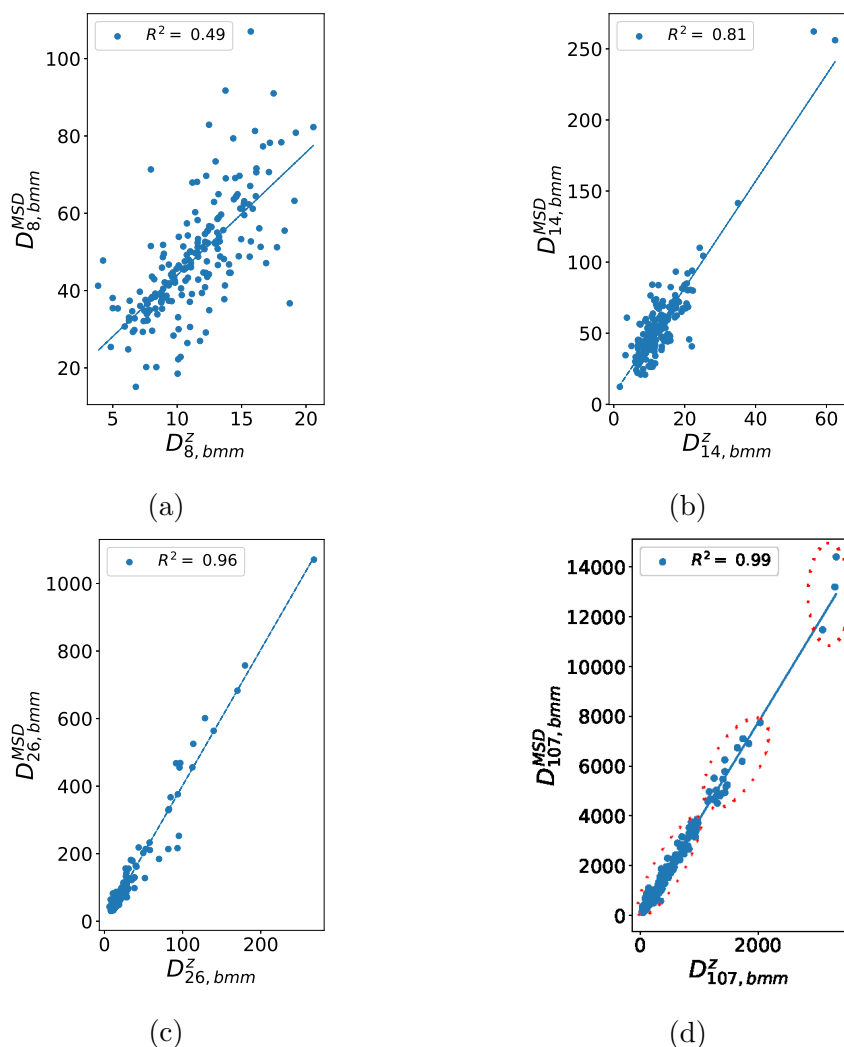


Figura 17: Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^z$ obtenidos con el algoritmo *bmm* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para ambas matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos. En la figura (d) se muestran con elipses discontinuas en color rojo tres regiones separadas de puntos de manera aproximada que podrían representar clústeres de las estructuras cristalinas de la formamida.

En las figuras 16 y 17 se puede observar que, conforme aumenta el número de moléculas que rodean a la molécula de referencia ($l=8, 14, 26, 107$) en las estructuras de formamida analizadas, la relación entre $\mathbf{D}^{MSD}$ y $\mathbf{D}^z$ se vuelve más lineal, independientemente del algoritmo empleado (*bmp* o *bmm*). Sin embargo, en la figura 15a hay una diferencia evidente entre estas métricas (*MSD* y *z*); asimismo, en las figuras 16a-16b y 17a-17b los valores de las matrices de distancia difieren considerablemente para el análisis basado en 8 y 14 moléculas vecinas. Por lo tanto, las discrepancias entre $\mathbf{D}^{MSD}$ y $\mathbf{D}^z$ se pierden conforme aumenta el parámetro l . Esto podría implicar que la comparación de los ambientes químicos alrededor de la molécula de referencia

dada por $\mathbf{D}^z$ mejora conforme aumenta l , ya que sus valores tienden a ser comparables con los de $\mathbf{D}^{MSD}$. Además, en los análisis con $l = 107$ es posible apreciar puntos que se separan de la región en la esquina inferior izquierda donde hay una mayor densidad de puntos como se indica en las figuras 16d y 17d. Esto implicaría también que se mejora la comparación de los ambientes químicos alrededor de la molécula de referencia al aumentar l , ya que considerando el mayor valor de moléculas vecinas ($l = 107$) es posible distinguir regiones de puntos en las figuras 16d y 17d cuyas estructuras correspondientes podrían separarse en tres clústeres como se indica en estas gráficas, similar al análisis realizado por Sadeghi et al. (2013) para otro tipo de métricas.

Sin embargo, una relación lineal entre $\mathbf{D}^{MSD}$ y $\mathbf{D}^z$ (figs. 16 y 17) puede implicar también la pérdida de información relevante de la comparación por pares de moléculas al aumentar el número de moléculas vecinas consideradas (l). En otras palabras, las métricas terminan adoptando valores tan grandes que se vuelven similares para todos los pares de estructuras y se vuelve difícil identificar aquellas que son diferentes. En términos del algoritmo *bmp*, esto se debe a que las sumas que condensan las matrices de comparación incrementan conforme aumenta l (ec. 107), mientras que para el algoritmo *bmm* simplemente se consideran valores de las métricas (en la matriz de comparación) cada vez más grandes (el máximo de los l mínimos, ec. 108). Esto también se refleja en la circunstancia que el coeficiente de determinación, R^2 , es mayor entre $\mathbf{D}^{MSD}$ y $\mathbf{D}_{14,bmm}^z$ que con $\mathbf{D}_{14,bmp}^z$ (figs. 16b y 17b); es decir, la relación lineal entre $\mathbf{D}^{MSD}$ y $\mathbf{D}_{bmm}^z$ es apreciable incluso a valores pequeños de l , ya que en general el algoritmo *bmm* lleva a valores más grandes de las métricas entre pares de estructuras cristalinas que *bmp*. Así, las peculiaridades identificadas por cada métrica se pierden al aumentar l . Esta última hipótesis es la más factible para explicar la relación lineal entre las matrices $\mathbf{D}^{MSD}$ y $\mathbf{D}^z$ para valores grandes de l , como se verá más adelante.

En el caso de la matriz $\mathbf{D}_{bmp}^{RCD}$ (fig. 18), la similitud con la medida $\mathbf{D}_{bmp}^{RMSD}$ también incrementa conforme el número de moléculas vecinas a la molécula de referencia aumenta. Esto reafirma la idea que al considerar un valor grande del parámetro l al construir las matrices de distancia, se podría perder información relevante sobre la comparación por pares de moléculas. Por lo tanto, el cálculo de las matrices de distancia impone otro filtro importante en la comparación de la configuración de moléculas en redes cristalinas, el cual puede o no preservar información relevante en torno a la comparación de las moléculas en un par de estructuras cristalinas (α y β). Esto evidentemente se verá reflejado en los clústeres generados a partir de cada matriz de distancia, como se analizará en las siguientes secciones. Aun así,

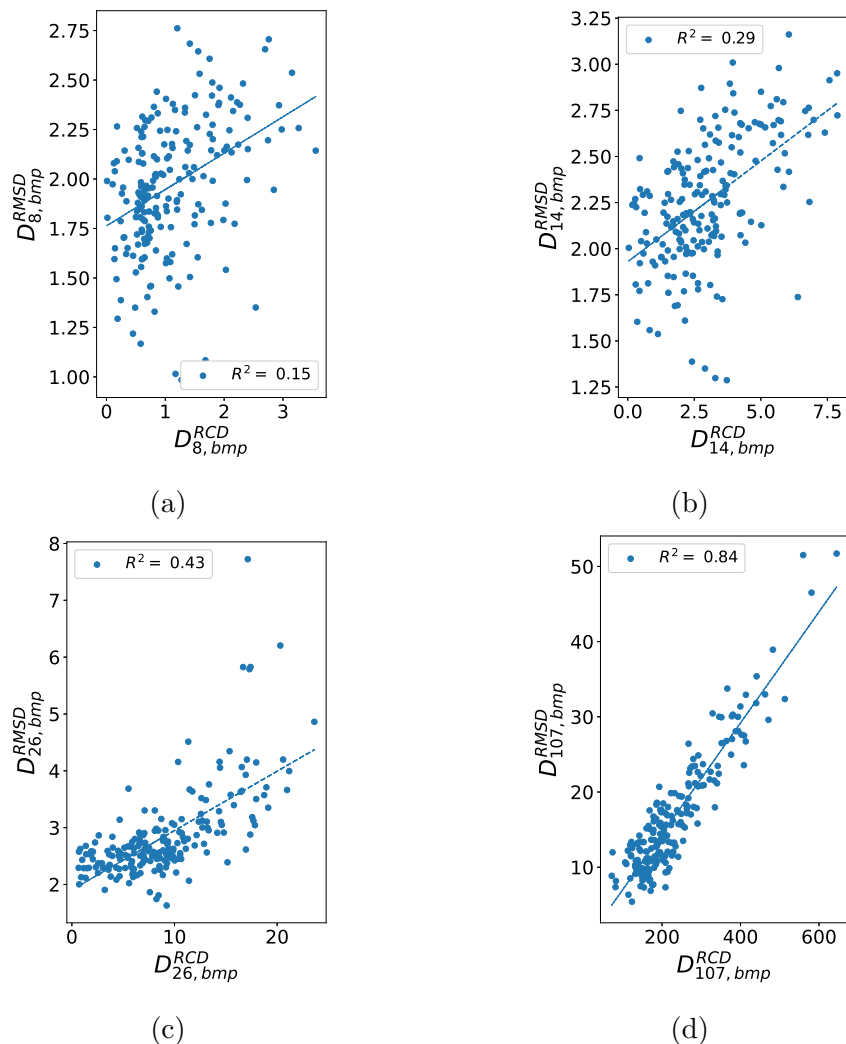


Figura 18: Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{RMSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^{RCD}$ obtenidos con el algoritmo *bmp* considerando (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia. Para ambas matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos.

es importante notar que el coeficiente R^2 entre $\mathbf{D}_{bmp}^{RCD}$ y $\mathbf{D}_{bmp}^{RMSD}$ no llega a ser tan alto como entre $\mathbf{D}^z$ y $\mathbf{D}^{MSD}$ (figs. 16 y 17). Por lo tanto, la correlación entre las métricas z y $RMSD$ sigue siendo mayor que entre RCD y $RMSD$, como se mostró previamente (figs. 15a y 15c).

Por último, en el caso de la medida $\mathbf{D}^s$ se observó el mismo comportamiento respecto a $\mathbf{D}^{RMSD}$ que se discutió para las otras matrices de distancia. Es decir, conforme aumenta el parámetro l mejora la correlación entre estas dos métricas. Sin embargo, de acuerdo con lo discutido en los párrafos anteriores, esto se debe más a un efecto de los métodos *bmp* y *bmm* en el cálculo de las matrices de distancia que a las propias métricas. Además, en general, se obtuvieron coeficientes de correlación

R^2 menores que con las otras métricas. Por lo tanto, se corroboró que la medida s no es adecuada para comparar los ambientes químicos alrededor de la molécula de referencia de redes cristalinas moleculares. Así, al comparar las matrices $\mathbf{D}^s$ y $\mathbf{D}^{RMSD}$, no se obtuvo información adicional sobre el comportamiento de s . Por lo tanto, las gráficas de este estudio se incluyen en el anexo A (figs. 32 y 33, pág. 161), con el fin de enfocar el análisis en las métricas que dieron los mejores resultados.

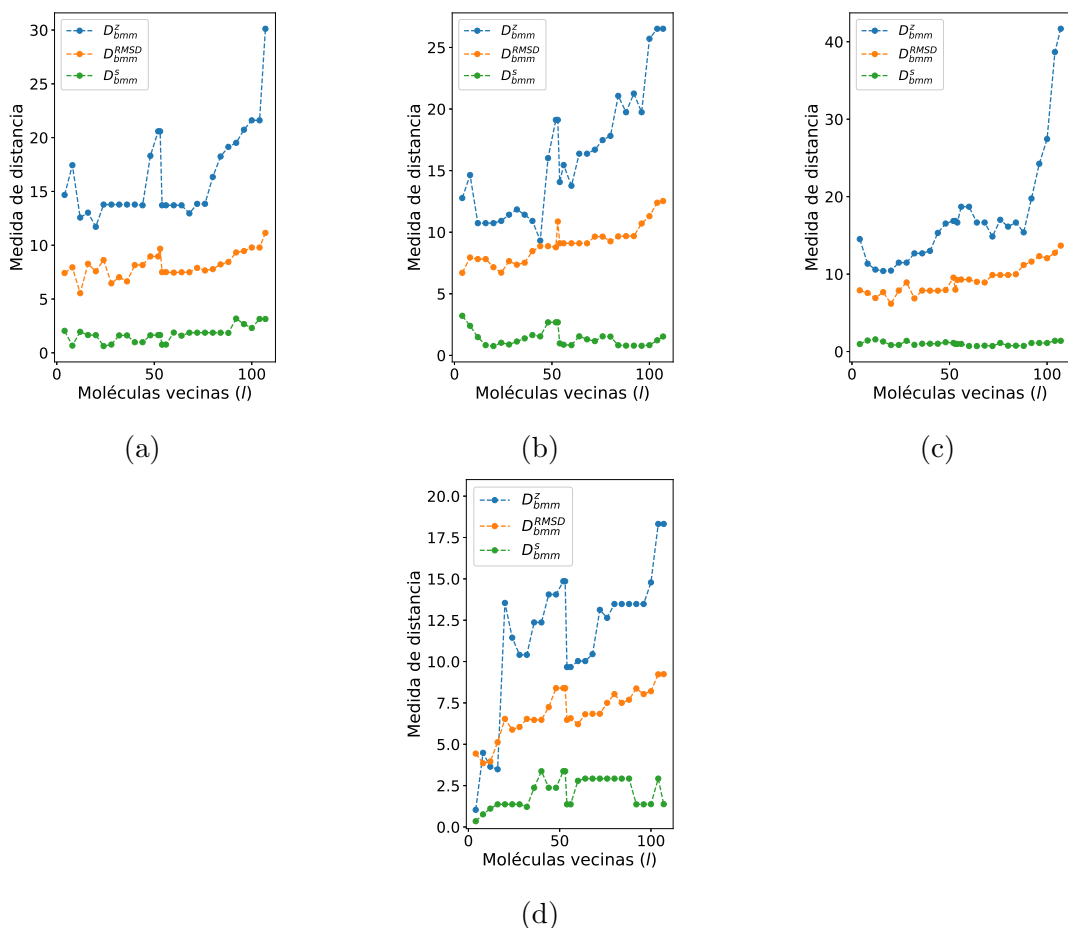


Figura 19: Métricas $\mathbf{D}^z$, $\mathbf{D}^s$ y $\mathbf{D}^{RMSD}$ entre los polimorfos (a) $MH1 - MH2$, (b) $MH1 - MH3$, (c) $MH2 - MH3$ y (d) $OA\alpha - OA\beta$ calculadas con el algoritmo bmm considerando diferentes valores de moléculas vecinas (l) en relación a la molécula de referencia. Para este análisis se emplearon los datos de las estructuras cristalinas experimentales reportados en la CSD (referencias en la tabla 1).

Por otra parte, las matrices de distancia ($\mathbf{D}^z$, $\mathbf{D}^s$, $\mathbf{D}^{RCD}_{bmp}$ y $\mathbf{D}^{RMSD}$) se aplicaron también para el análisis de los polimorfos de la hidrazida maleica (tres polimorfos) y del ácido oxálico (dos polimorfos), con el fin de identificar si estas métricas son aptas para permitir la distinción entre las estructuras cristalinas de un mismo compuesto. Los resultados muestran que todas las métricas $\mathbf{D}$ son capaces de diferenciar entre los polimorfos, a excepción de la medida $\mathbf{D}^s_{bmp}$ (figs. 19 y 20). En particular, las diferencias entre los polimorfos son exacerbadas con el algoritmo bmm (fig. 19), lo

cual cumple con el objetivo de este método, ya que este busca la mayor diferencia posible para cada par de estructuras. Sin embargo, los valores de las matrices de distancia con *bmm* fluctúan respecto al valor de l . Aunque existen rangos de valores de l para los cuales estas métricas se mantienen constantes, estas regiones varían para las diferentes estructuras cristalinas y para las diferentes métricas. Así, esto también complica la elección del número de moléculas cercanas (l) óptimo en la comparación de estructuras cristalinas.

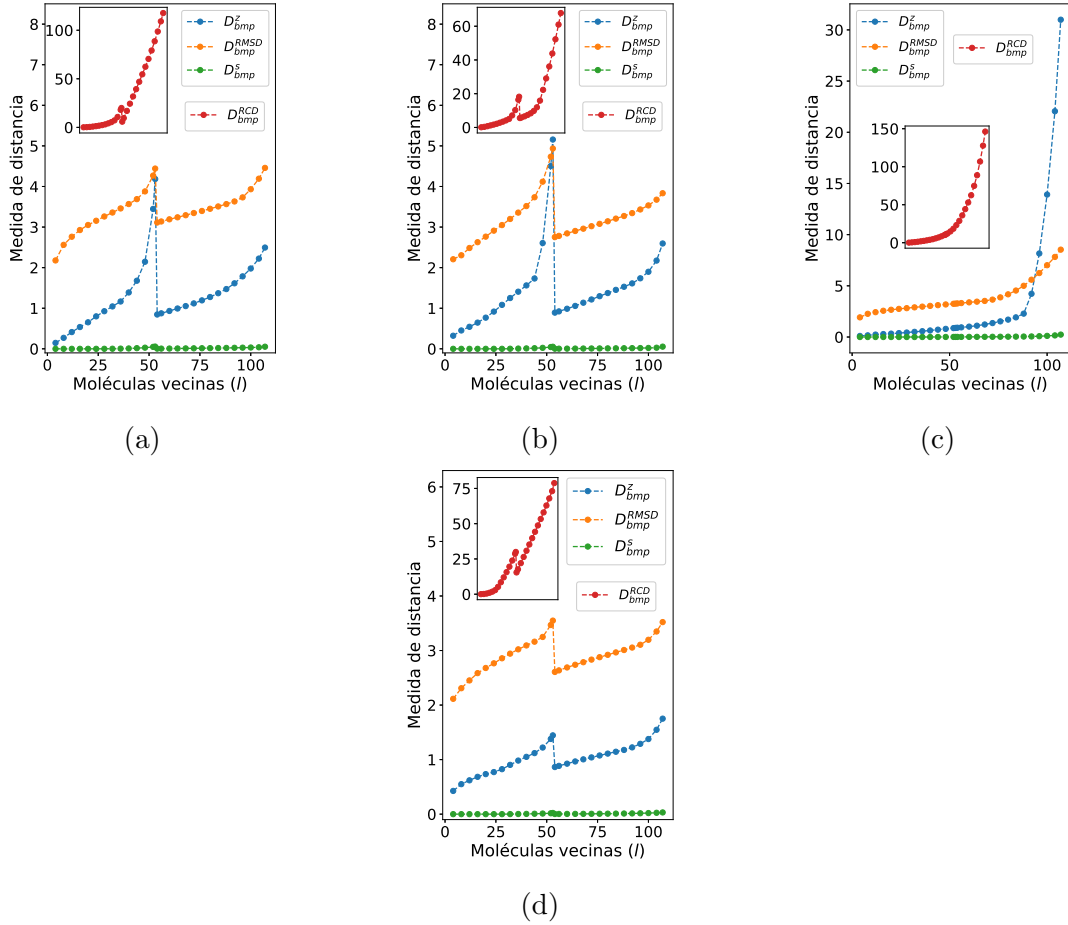


Figura 20: Métricas $\mathbf{D}^z$, $\mathbf{D}^s$ y $\mathbf{D}^{RMSD}$ entre los polimorfos (a) $MH1 - MH2$, (b) $MH1 - MH3$, (c) $MH2 - MH3$ y (d) $OA\alpha - OA\beta$ calculadas con el algoritmo *bmp* considerando diferentes valores de moléculas vecinas (l) en relación a la molécula de referencia. Para este análisis se emplearon los datos de las estructuras cristalinas experimentales reportados en la CSD (referencias en la tabla 1). Los valores de la matriz $\mathbf{D}_{bmp}^{RCD}$ se presentan de manera separada debido a que su escala es mucho mayor a la del resto de las métricas.

En el caso del algoritmo *bmp*, se observa una tendencia creciente de las métricas $\mathbf{D}_{bmp}^z$, $\mathbf{D}_{bmp}^{RMSD}$ y $\mathbf{D}_{bmp}^{RCD}$ conforme aumenta el valor del parámetro l (fig. 20). Esto concuerda con la metodología implementada, ya que al aumentar el número de moléculas cercanas a la molécula de referencia (l) los mínimos simplemente se suman a los mínimos previos (ec. 107), tal que aumenta el valor de la distancia entre las

configuraciones de los pares de estructuras. Sin embargo, se observa una discontinuidad importante al usar este algoritmo. Esto se debe a que al comparar estructuras con un número diferente de moléculas en la celda unitaria, como *MH1* ($Z = 2$) y *MH2* ($Z = 4$), el número de cuaterniones duales generados para cada estructura es diferente. En el caso de la estructura con 2 moléculas en la celda unitaria (*MH1*), para $l < 53$ solo es necesario considerar la primera zona periódica para generar al menos 53 moléculas; mientras que para valores de $l > 53$ es necesario considerar la segunda zona periódica, lo cual incrementa el número de moléculas a 206. Para *MH2* ($Z = 4$), se generan 107 moléculas (sin considerar la molécula de referencia) en todo el rango de valores de l . Así, las matrices de comparación son distintas. Por lo tanto, resulta importante considerar este límite ($l = 53$) en la interpretación de los análisis. Evidentemente, en el caso de comparar estructuras con el mismo número de moléculas en la celda unitaria, como el conjunto de datos de la formamida, esta discontinuidad o cambio en las matrices de comparación no se presenta. Asimismo, esta discontinuidad se presenta también en los análisis con el algoritmo *bmm* (fig. 19), aunque es menos evidente debido al resto de las fluctuaciones observadas.

Por lo tanto, ambos algoritmos de búsqueda de mínimos (*bmp* y *bmm*) generan cambios importantes en las métricas en función del valor del parámetro l elegido. Por ejemplo, en el caso de la hidrazida maleica con el algoritmo *bmp* (figs. 20a a 20c), la métrica $\mathbf{D}_{bmp}^z$ indica para la mayoría de los valores de l que *MH2* y *MH3* son similares entre sí, pero diferentes de *MH1*; sin embargo, si se consideran 107 moléculas vecinas, entonces la misma métrica indica lo contrario. Además, se observa un cambio abrupto en los valores de la medida $\mathbf{D}_{bmp}^z$ entre *MH2* y *MH3* (fig. 20c) en la región cercana a 107 moléculas vecinas, lo cual no se presenta en ninguna de las otras comparaciones. Por lo tanto, la manera en que las métricas dependen del parámetro l puede cambiar de la comparación de un par de estructuras a otro.

Por otra parte, cabe destacar que la matriz de distancia $\mathbf{D}^s$ es en la mayoría de los casos indistinta ante las diferentes estructuras cristalinas comparadas (figs. 19 y 20). Solo en el caso del algoritmo *bmm* (fig. 19) la medida $\mathbf{D}^s$ muestra diferencias entre los polimorfos, en particular en la comparación de las estructuras cristalinas del ácido oxálico (fig. 19d). Sin embargo, esto se debe al algoritmo *bmm* que exagera los valores de todas las métricas. Además, los valores de la matriz $\mathbf{D}_{bmm}^s$ se encuentran por debajo de los del resto de matrices. Por lo tanto, esto es otra prueba de que la medida s no genera una adecuada descripción del espacio configuracional de redes cristalinas moleculares. Así, se decidió descartar esta medida en los análisis subsecuentes.

En resumen, se mostró que la medida de disimilitud de pares de moléculas z

muestra un comportamiento similar al MSD , lo cual es un indicio de que esta métrica (z) genera una descripción adecuada del espacio configuracional. Por lo contrario, la medida s no fue capaz de distinguir de manera importante entre las diferentes orientaciones de las moléculas, tal que se decidió descartar esta métrica. Por otra parte, al calcular las matrices de distancia se observó que es posible que se pierda información en relación a la comparación de pares de moléculas importante. Así, las similitudes entre las diferentes matrices de distancia dependieron del método usado (bmp o bmm) y del parámetro l . Asimismo, se observaron cambios en las tendencias de las métricas ($\mathbf{D}$) conforme aumenta l en la comparación entre los polimorfos $MH2$ y $MH3$, respecto al resto de las comparaciones. En consecuencia, la variación de las métricas en función del parámetro l puede ser considerablemente diferente para algunos pares de estructuras cristalinas. Por lo tanto, esto afectará los clústeres generados con cada una de las matrices de distancia, como se presenta en la siguiente sección.

6.4. Aplicación del método de agrupamiento espectral

De acuerdo con el análisis en la sección previa, z permite generar comparaciones entre las posiciones y orientaciones de pares de moléculas de manera similar al $RMSD$ (fig. 15a). Así, con el fin de continuar con la validación de esta métrica se generaron clústeres de estructuras cristalinas de la formamida a partir del conjunto de 15594 estructuras generadas aleatoriamente y con diferentes matrices de distancia de z , así como del $RMSD$ y del RCD , empleando el método de agrupamiento espectral. Estos clústeres representan conjuntos de estructuras cristalinas con configuraciones (arreglos espaciales de las moléculas en las redes cristalinas) similares, de acuerdo con los datos de la matriz de distancia empleada en el agrupamiento. En esta sección, se estudian los cambios que resulten en los clústeres, a través del coeficiente de silueta promedio, al variar el número de moléculas cercanas a la molécula de referencia (l) en el cálculo de las matrices $\mathbf{D}$, y el número de estructuras cristalinas analizadas para un mismo compuesto. Además, se analizan las distribuciones de energía en los clústeres generados y el $CV(RMSD)$ (ec. 111) de las distancias de las estructuras a sus *medoides* correspondientes. El objetivo de esto es analizar la posibilidad de que los clústeres generados permitan identificar estructuras relevantes, aquellas con configuraciones similares a las de varias estructuras, lo cual pueda aplicarse como un filtro de información en metodologías para la predicción de la estructura cristalina de compuestos moleculares. Cabe mencionar que en la primera parte de esta sección se mostrará el análisis de un subconjunto de 3000 estructuras cristalinas de la formamida, ya que la realización de este análisis sobre el conjunto completo de estructuras (15594) no fue computacionalmente factible debido a la gran cantidad de matrices de distancia calculadas. Sin embargo, se muestra posteriormente el análisis del conjunto completo de estructuras (15594) con un menor número de matrices de distancia, con base en los resultados del análisis del subconjunto mencionado.

Por otra parte, cabe mencionar que la métrica z_{rc} fue descartada debido a que se obtuvieron mejores clústeres (mayores valores del coeficiente de silueta promedio) empleando simplemente la métrica z . Por lo tanto, no se tiene un análisis extenso de z_{rc} , aunque el análisis debido al cual se descartó esta medida se presenta en el anexo A (fig. 34, pág. 163). Esto último con el fin de simplificar y enfocar el siguiente estudio en solo una de las métricas propuestas: z .

En el caso de las medidas $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RMSD}$, se puede observar en la figura 21a que al considerar un mayor número de moléculas cercanas a la molécula de referencia (l) se generan mejores agrupaciones; es decir, valores mayores del coeficiente de silueta promedio. Sin embargo, la tendencia no es monótonamente creciente en el caso de estas matrices de distancia, ya que se observan un mínimo (en $l = 60$) y dos máximos

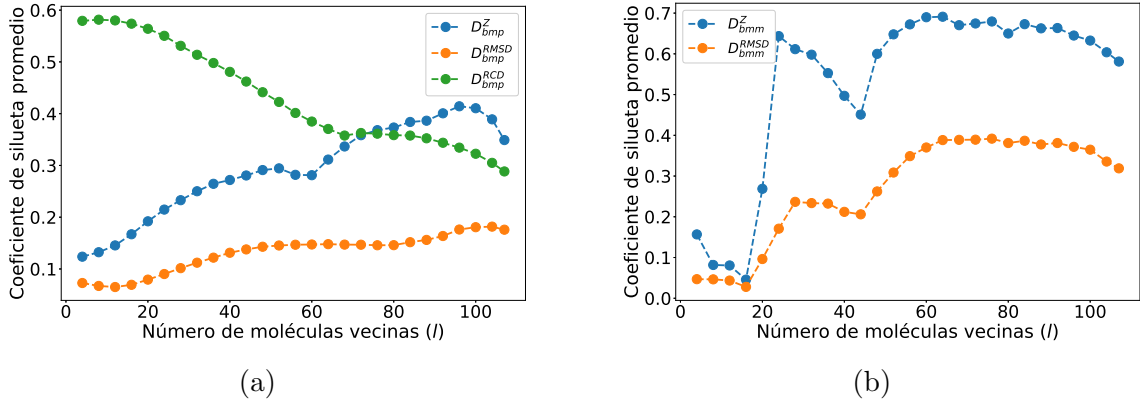


Figura 21: Coeficientes de silueta promedio de las agrupaciones obtenidas con las métricas z , $RMSD$ y RCD para números diferentes de moléculas vecinas (l) en relación a la molécula de referencia, considerados al calcular las matrices de distancia con los algoritmos (a) bmp y (b) bmm para un subconjunto de 3000 estructuras cristalinas de la formamida; en todos los casos se usó el método de agrupamiento espectral.

(en $l = 52$ y $l = 96$) locales en el caso de $\mathbf{D}_{bmp}^z$ y un mínimo (en $l = 12$) y un máximo (en $l = 104$) locales en el caso de $\mathbf{D}^{RMSD}$. Además, resulta importante remarcar que el puntaje de silueta, al depender de las matrices de distancia empleadas, no aporta información sobre la validez de la comparación de la configuración de las estructuras generada con estas métricas. Es decir, aunque el coeficiente de silueta promedio muestra de forma correcta los clústeres óptimos, cada uno de estos clústeres puede no necesariamente representar estructuras con configuraciones similares si existen errores en las métricas empleadas. Por lo tanto, con el fin de contrastar e interpretar las agrupaciones identificadas como óptimas, se presentarán, por una parte, los análisis que consideran 107 moléculas cercanas y, por otra parte, también análisis con valores de l menores que no necesariamente generan los valores más altos del puntaje de silueta promedio para $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RMSD}$. Evidentemente, el valor de $l = 107$ no genera el coeficiente de silueta promedio máximo para ninguna de estas matrices (fig. 21a); sin embargo, sí genera valores cercanos al máximo en cada caso. Además, este valor de l (107) es el número total de moléculas alrededor de la molécula de referencia (M) en la primera zona periódica ($z_p=1$) para las estructuras de la formamida; por lo tanto, emplear este valor permite conservar la mayor cantidad de información al comparar cada par de estructuras cristalinas. Asimismo, las posiciones de los máximos observados en la figura 21a podrían variar al analizar un mayor número de estructuras cristalinas de la formamida; en cuyo caso sería necesario calcular las matrices de distancia ($\mathbf{D}^z$, $\mathbf{D}^{RMSD}$ y $\mathbf{D}_{bmp}^{RCD}$) considerando diferentes valores del parámetro l y generar gráficas similares a las presentadas en la figura 21 para los subconjuntos más grandes de estructuras cristalinas de la formamida, lo

cual sería computacionalmente muy costoso. Por lo tanto, se optó por considerar el valor (en la primera zona periódica) de l máximo (107) como el representativo para los mejores clústeres.

A diferencia de los valores del coeficiente de silueta promedio obtenidos para $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RMSD}$, el coeficiente de silueta promedio de la medida $\mathbf{D}_{bmp}^{RCD}$ muestra una tendencia más homogénea y decreciente conforme aumenta el valor de l (fig. 21a). En este caso es evidente que las mejores agrupaciones se obtienen al considerar 8 moléculas cercanas a la molécula de referencia, lo cual concuerda con que $l = 8$ es el parámetro estándar para esta métrica (Li et al. 2018). Por lo tanto, en el caso del *RCD* se enfocará el análisis en la matriz $\mathbf{D}_{8,bmp}^{RCD}$, en esta sección. El hecho de que la tendencia del puntaje de silueta promedio para $\mathbf{D}_{bmp}^{RCD}$ sea contraria a las observadas con $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RMSD}$ reafirma que el *RCD* genera un muestreo del espacio configuracional diferente al de las otras dos métricas, principalmente para valores pequeños de l , como se mostró en las figuras 16 a 18.

También en el caso del algoritmo *bmm* (fig. 21b) se observan mejores agrupaciones para valores grandes de l . En particular, para valores de l mayores a 60 se observa que el coeficiente de silueta promedio se mantiene casi constante, lo cual implica que cualquier valor de l en este rango puede generar buenas agrupaciones. Sin embargo, para valores de l menores a 60 el coeficiente de silueta promedio fluctúa considerablemente, tal que resulta difícil seleccionar en este intervalo un valor de l que asegure buenos resultados. Así, como se mencionó para las medidas $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RMSD}$, también se analizarán para *bmm* las agrupaciones resultantes con diferentes valores de l con el fin de contrastar los diferentes tipos de clústeres generados. Además, cabe mencionar que las fluctuaciones observadas se deben en parte al tipo de búsqueda de mínimos realizada con *bmm*, ya que para esta se modifican las matrices de comparación; tal que no existe una correlación clara entre los mínimos encontrados conforme aumenta l , contrario al caso de *bmp*. Por otra parte, los valores del puntaje de silueta promedio obtenidos con el algoritmo *bmm* son mayores a los obtenidos con *bmp*, lo cual implica que el primer método genera mejores agrupaciones. Sin embargo, aún es necesario detallar el análisis de los clústeres con el fin de comprender lo que los clústeres óptimos representan en términos de las configuraciones de las estructuras cristalinas que los conforman.

Además, se puede observar en la figura 22a que las medidas $\mathbf{D}_{bmp}^{RCD}$ y $\mathbf{D}_{bmp}^z$ permiten identificar diferentes números de clústeres óptimos según el número de moléculas alrededor de la molécula de referencia (l) considerado, mientras que con la medida $\mathbf{D}_{bmp}^{RMSD}$ se identifican dos clústeres en todos los casos. Por una parte, esto muestra, como se había comentado en los análisis anteriores, que variaciones del parámetro l

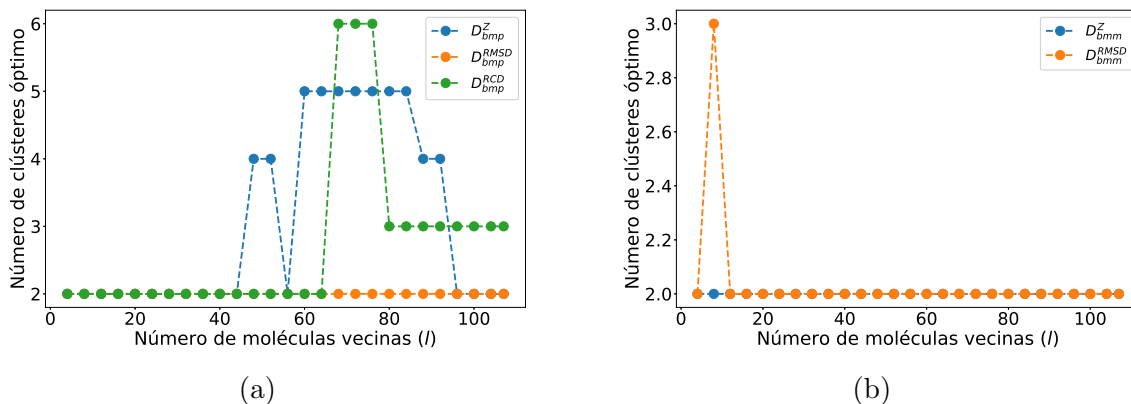


Figura 22: Números de clústeres óptimos (generan los valores más altos del puntaje de silueta promedio) obtenidos con las métricas z , $RMSD$ y RCD para diferentes valores de moléculas cercanas (l) en relación a la molécula de referencia, considerados al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un subconjunto de 3000 estructuras de la formamida; en todos los casos se usó el método de agrupamiento espectral.

llevan a muestreos del espacio configuracional diferentes y, por lo tanto, la cantidad de clústeres necesaria para asociar (agrupar) las estructuras cristalinas con configuraciones similares varía. Por otra parte, en el caso de la matriz $\mathbf{D}^z_{bmp}$ existe un contraste importante entre el número óptimo de clústeres (fig. 22a) y el coeficiente de silueta promedio (fig. 21a). En el rango de aproximadamente 80 a 100 moléculas alrededor de la molécula de referencia (l) donde el coeficiente de silueta promedio alcanza valores cercanos al máximo, se observan cambios importantes en el número de clústeres. Los clústeres generados podrían emplearse, por ejemplo, para reducir el análisis de las 3000 estructuras cristalinas de la formamida al análisis de solo las estructuras relevantes en cada clúster, las cuales pueden ser las estructuras con menor energía o los *medoides*. Entonces, pequeños cambios de l pueden llevar a generar números de clústeres de las estructuras cristalinas muy distintos y, en consecuencia, a identificar estructuras cristalinas relevantes diferentes. Esto evidentemente representa un problema serio en la elección del parámetro l al calcular las matrices de distancia; por lo tanto, los siguientes análisis se enfocan en comprender el efecto de l en los clústeres generados con el fin de delimitar el problema.

En el caso del algoritmo *bmm*, este resulta en dos clústeres con las medidas z y $RMSD$ para todos los valores del parámetro l , excepto para $l = 8$ con $RMSD$ (fig. 22b)⁶. Así, en este caso, las variaciones del coeficiente de silueta promedio

⁶Este cambio en el número de clústeres puede deberse a una distribución particular de los datos en la matriz $\mathbf{D}^{RMSD}_{8,bmm}$, en comparación con el resto de las matrices; sin embargo, no se realizó un análisis detallado de esta diferencia, ya que se priorizaron otros análisis que permitiesen cumplir con el objetivo de este proyecto: validar si es posible identificar estructuras con configuraciones similares a partir de los clústeres generados con el método de agrupamiento.

se deben a que la asignación de las estructuras cristalinas a cada uno de los dos clústeres cambia al modificar el valor de l y no a un cambio en el número de clústeres generados, como en el caso de las matrices calculadas con el algoritmo *bmp*. En otras palabras, el número de clústeres necesarios (2) para separar de manera óptima las 3000 estructuras cristalinas se mantiene; sin embargo, es evidente que en el caso del algoritmo *bmm* las estructuras cristalinas que se consideran similares (en términos de sus configuraciones) también dependen del parámetro l . En consecuencia, los resultados para ambos algoritmos (*bmp* y *bmm*) reafirman la necesidad de considerar múltiples valores de l en los análisis subsecuentes. En consecuencia, se optó por analizar las matrices de distancia con el parámetro l igual a 14, 26 y 107, tratando de evitar la arbitrariedad en la elección de este, de acuerdo con lo descrito en la metodología (cf. sección 5.4). Por lo tanto, los análisis posteriores estarán centrados en estos valores de l para las matrices de distancia.

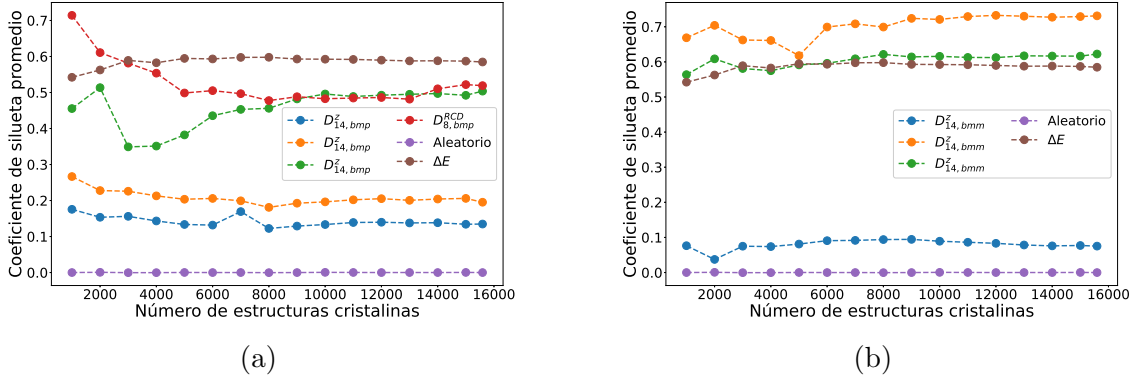


Figura 23: Coeficientes de silueta promedio óptimos de los clústeres generados con las métricas z y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para subconjuntos de estructuras cristalinas de la formamida de diferente tamaño; en todos los casos se usó el método de agrupamiento espectral. Además, en morado se muestran los valores correspondientes a generar las agrupaciones de manera aleatoria considerando solo 2 clústeres y calcular el coeficiente de silueta con la matriz $\mathbf{D}_{107,bmp}^z$; en café se muestran los valores para las agrupaciones obtenidas considerando la diferencia de energía electrónica entre todos los pares de estructuras cristalinas del conjunto analizado ($\Delta E_{\alpha,\beta} = |E^\alpha - E^\beta| \text{ kcal/mol}$).

Por otra parte, se estudió el comportamiento del agrupamiento respecto al número de estructuras cristalinas analizadas con el fin de evaluar si las conclusiones establecidas para el subconjunto de 3000 estructuras cristalinas siguen siendo válidas al considerar un mayor número de estructuras. Asimismo, se agruparon las estructuras cristalinas de la formamida de manera aleatoria en dos clústeres (fig. 23), como un caso contrario a agrupar las estructuras con base en la similitud entre sus configuraciones (aplicación del método de agrupamiento espectral) y dado que la mayoría de las matrices de distancia generaron dos clústeres. En este caso, se emplearon los

datos de la matriz $\mathbf{D}_{107,bmp}^z$, que se seleccionó como un ejemplo de las matrices de z , en conjunto con las etiquetas (asignación de las estructuras a cada clúster) generadas de manera aleatoria para calcular los coeficientes de silueta (ec. 56). Pero la matriz $\mathbf{D}_{107,bmp}^z$ no se empleó para generar los clústeres como en el resto de los análisis presentados en la figura 23. Además, se estudiaron los clústeres que fueron generados considerando la diferencia de energía entre pares de estructuras cristalinas (ΔE , en $kcal/mol$); esto como un caso de una distancia simple que permite contrastar el resto de las métricas ($\mathbf{D}^z$ y $\mathbf{D}_{bmp}^{RCD}$) y con el fin de analizar la distribución de energías electrónicas de las redes cristalinas. Cabe mencionar que en los siguientes análisis, no se consideró la medida *RMSD* debido a que esta es computacionalmente costosa y no fue factible realizar el análisis con ella para más de 3000 estructuras cristalinas, aunque esta métrica se retomará en el análisis de las redes cristalinas del ácido oxálico y de la hidrazida maleica.

En el caso del algoritmo *bmp* para los subconjuntos más pequeños (hasta aproximadamente 8000 estructuras cristalinas, fig. 23a), para todas las matrices de distancia $\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmp}^{RCD}$ existen cambios importantes en el coeficiente de silueta promedio y, por ende, en los clústeres generados. En cambio, para los subconjuntos de mayor tamaño el coeficiente de silueta promedio se mantiene casi constante. Esto implica que la separación óptima de las estructuras cristalinas en clústeres deja de depender de la cantidad de estructuras que se analice, una vez que se tiene una cantidad de estructuras lo suficientemente representativa. Los valores del coeficiente de silueta promedio para las matrices de z y de *RCD* convergen de manera similar, lo cual implica que ambas métricas muestran que no existen cambios importantes en las configuraciones de las estructuras cristalinas a partir de que el subconjunto consiste de 8000 o más redes cristalinas. Es decir, las estructuras cristalinas de la formamida que se “agregan” al aumentar el tamaño de los subconjuntos a partir de 8000 estructuras tienen configuraciones similares a algunas de las estructuras que se encuentran en los subconjuntos de menor tamaño. Por lo tanto, los subconjuntos de más de 8000 estructuras resultan en clústeres similares, tal que el coeficiente de silueta promedio se mantiene casi constante. Esto último tomando en cuenta que el número de clústeres tampoco se ve afectado por el tamaño del subconjunto de estructuras cristalinas, ya que para todas las matrices $\mathbf{D}$ con *bmp* se generaron dos clústeres; excepto con la matriz $\mathbf{D}_{26,bmp}^z$ para el conjunto completo (15594 estructuras), como se mostrará más adelante.

Por el contrario, en el caso del algoritmo *bmm* (fig. 23b), no se observan cambios importantes del coeficiente de silueta promedio en todo el intervalo de número de estructuras cristalinas, excepto para la matriz $\mathbf{D}_{26,bmm}^z$. Por lo tanto, este algoritmo

es menos sensible al tamaño del conjunto de estructuras cristalinas considerado y, por ende, a cambios de las configuraciones, comparado con *bmp*. Esto reafirma que los algoritmos de búsqueda de mínimos pueden llevar a descripciones del espacio configuracional considerablemente distintas. Por otra parte, los valores del coeficiente de silueta promedio para $\mathbf{D}_{26,bmm}^z$ y $\mathbf{D}_{107,bmm}^z$ (fig. 23b) son mayores a los obtenidos con cualquier matriz de distancia con el algoritmo *bmp* (fig. 23a). Así, considerando únicamente el puntaje de silueta promedio, estas matrices de distancia generan los mejores clústeres sin importar el tamaño del subconjunto de estructuras analizado.

Cabe remarcar que en todos los casos se obtuvieron valores del coeficiente de silueta promedio superiores a los que se obtienen al asignar los clústeres de manera aleatoria (fig. 23). Por lo tanto, todas las métricas generan agrupaciones significativas. Asimismo, los valores del coeficiente de silueta promedio para cualquiera de las métricas en el espacio configuracional muestran tendencias similares a las obtenidas con las matrices que involucran diferencias de energía ($\mathbf{D}^{\Delta E}$, fig. 23). Esta última es una comparación directa de las estructuras cristalinas, a diferencia de las métricas en el espacio configuracional, ya que la energía es un descriptor fundamental de estos sistemas y la diferencia absoluta de las energías es una métrica bien establecida. Además, $\mathbf{D}^{\Delta E}$ genera puntajes de silueta promedio superiores a 0.5 para todos los subconjuntos de estructuras analizados, lo cual muestra que esta métrica permite generar buenas agrupaciones. Por lo tanto, las métricas $\mathbf{D}^z$ (con *bmp* o *bmm*) y $\mathbf{D}_{bmp}^{RCD}$ muestran un comportamiento similar a una medida más simple y bien establecida ($\mathbf{D}^{\Delta E}$), lo cual reafirma la idea de que *z* y *RCD* son en general adecuadas como medidas de disimilitud (métricas). Sin embargo, cabe recordar que aunque las métricas en el espacio configuracional permiten generar clústeres de manera adecuada, aún es necesario analizar en qué medida estos clústeres representan estructuras cristalinas con configuraciones similares.

En los casos de $\mathbf{D}_{14,bmp}^z$, $\mathbf{D}_{16,bmp}^z$ y $\mathbf{D}_{14,bmm}^z$ los valores del puntaje de silueta promedio son cercanos a cero, lo que implica que no evidencian una correlación importante entre las estructuras agrupadas con estas matrices o que simplemente las medidas fallan en identificar similitudes configuracionales. Esto muestra que las tendencias observadas respecto a la variación de *l* para el subconjunto de 3000 estructuras cristalinas (fig. 21) se mantienen para subconjuntos de estructuras más grandes. Esto es, conforme aumenta el valor de *l* al calcular la matriz $\mathbf{D}^z$, con el algoritmo *bmp* o *bmm*, incrementan los valores del coeficiente de silueta promedio. Por lo tanto, los clústeres que resultan del agrupamiento dependen más del valor de *l* que del número de estructuras cristalinas agrupadas, siempre y cuando el conjunto sea lo suficientemente representativo, como el subconjunto de 8000 estructuras cris-

talinas. Aun así, con el fin de complementar el estudio de las estructuras cristalinas de la formamida, los siguientes análisis de las matrices de distancia se mostrarán para el conjunto completo de estructuras cristalinas (15594).

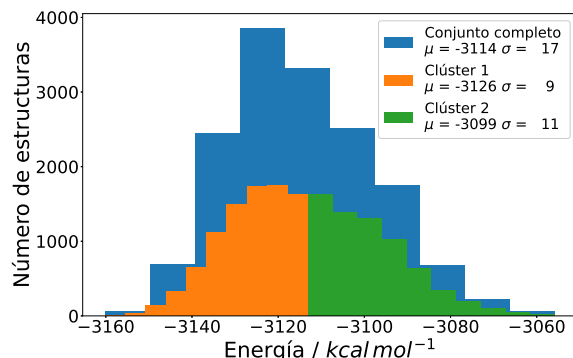


Figura 24: Histogramas de las energías de las estructuras cristalinas generadas aleatoriamente para la formamida. El histograma en azul considera la energía electrónica de todas las estructuras cristalinas (15594); en naranja y verde se muestran las energías de los dos clústeres de estructuras identificados con la matriz $\mathbf{D}^{\Delta E}$. Además, para cada caso se muestra el valor promedio, μ , y la desviación estándar, σ , en $kcal/mol$.

Clúster	$E_{min} / kcalmol^{-1}$	$\bar{E} / kcalmol^{-1}$	$\sigma(E) / kcalmol^{-1}$	PR (%)
1	-3,160.13	-3,126.27	8.61	57
2	-3,113.13	-3,098.49	10.64	43

Tabla 4: Análisis energético de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida, que se generaron aleatoriamente, con la matriz de distancia $\mathbf{D}^{\Delta E}$. Para cada clúster se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas; y la población relativa (PR).

Es importante remarcar que aunque los clústeres generados con la matriz $\mathbf{D}^{\Delta E}$ resultan en valores del coeficiente de silueta promedio altos (fig. 21), la desviación estándar de las energías en cada clúster es aún relativamente alta, considerando que una de las perspectivas de este proyecto es la predicción de la estructura cristalina de compuestos moleculares y que la diferencia de energía electrónica entre polimorfos es en promedio menor a 1 $kcal/mol$ (Cruz-Cabeza et al. 2015, Nyman y Day 2015). En la figura 24 y en la tabla 4 se puede observar que las desviaciones estándar de las energías para los dos clústeres son superiores a 8 $kcal/mol$, lo cual implica que cada clúster contiene estructuras cristalinas con energías significativamente distintas entre sí. No obstante, las desviaciones estándar de las energías se podrían reducir al generar más clústeres, mas esto difiere del objetivo de identificar el número de clústeres relevantes de manera automática. Asimismo, se podrían optimizar las geometrías moleculares y los parámetros de red de las estructuras cristalinas en cada

clúster, tal que podrían descartarse duplicados (estructuras cristalinas que convergen al mismo mínimo de energía), pero esto resulta computacionalmente costoso debido a la cantidad de estructuras consideradas. Por lo tanto, los clústeres obtenidos con $\mathbf{D}^{\Delta E}$ son un claro ejemplo de que la energía electrónica no es un criterio suficiente para la predicción de estructuras cristalinas partiendo de un conjunto vasto de estructuras generadas aleatoriamente y, por lo tanto, nace la necesidad de complementar el análisis con métricas en el espacio configuracional. Tal que, por ejemplo, estas métricas permitan identificar estructuras cristalinas representativas (son similares a varias estructuras con base en sus configuraciones), tal que se puedan realizar cálculos de estructura electrónica más precisos solo de las estructuras cristalinas relevantes. Sin embargo, como se mencionó previamente, la separación de estructuras cristalinas lograda en términos de la energía electrónica es evidente, ya que las desviaciones estándar que definen a los dos clústeres son menores a la del conjunto completo y no existen valores de energía que sean comunes a ambos clústeres; es decir, no existe una intersección entre los rangos de valores de energía $[\bar{E} - \sigma(E), \bar{E} + \sigma(E)]$ (fig. 24 y tab. 4). Esto permite reafirmar que el algoritmo de agrupamiento espectral empleado (fig. 10) funciona correctamente y que variaciones en las agrupaciones se deben principalmente a la matriz de distancia utilizada. Además, esto confirma, como se ha mencionado previamente, que la matriz $\mathbf{D}^{\Delta E}$ es una métrica correcta que sirve como referencia al analizar las métricas $\mathbf{D}^z$ y $\mathbf{D}_{bmp}^{RCD}$.

l	Clúster	$E_{min} / kcalmol^{-1}$	$\bar{E} / kcalmol^{-1}$	$\sigma(E)$	PR (%)	CV(RMSD)
14	1	-3,158.72	-3119.79	13.41	60	1.06
	2	-3,160.13	-3106.26	17.86	40	1.08
26	1	-3,158.72	-3114.51	16.39	64	1.35
	2	-3,160.13	-3120.14	13.69	25	1.22
	3	-3,157.00	-3099.87	16.57	11	1.42
107	1	-3,160.13	-3116.11	16.20	88	1.24
	2	-3,150.86	-3101.78	14.95	12	1.08

Tabla 5: Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con las matrices de distancia D_{bmp}^z que fueron calculadas con 14, 26 y 107 moléculas cercanas (l) en relación a la molécula de referencia. Para los clústeres obtenidos se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los *medoides* ($CV(RMSD)$).

En general, los clústeres de estructuras cristalinas obtenidos con las matrices de distancia $\mathbf{D}_{bmp}^z$, $\mathbf{D}_{bmm}^z$ y $\mathbf{D}_{bmp}^{RCD}$ contienen distribuciones de energía electrónica similares (tabs. 5 a 7). Así, los clústeres de estructuras cristalinas que fueron identificados como diferentes con base en las configuraciones de las estructuras cristalinas que

los conforman no son necesariamente distintos en términos de las distribuciones de energía electrónica. Además, las desviaciones estándar de las energías de los clústeres generados con las matrices de distancia $\mathbf{D}^z$ y $\mathbf{D}_{bmp}^{RCD}$ (tabs. 5 a 7) son mayores a las desviaciones estándar de energía establecidas con la medida $\mathbf{D}^{\Delta E}$ (tab. 4); incluso, estas llegan a ser similares a la desviación estándar de energía electrónica para el conjunto de estructuras cristalinas completo (17 kcal/mol). Por lo tanto, ninguna métrica utilizada en esta tesis ($\mathbf{D}_{bmp}^z$, $\mathbf{D}_{bmm}^z$ o $\mathbf{D}_{bmp}^{RCD}$) es capaz de correlacionar las configuraciones moleculares de las redes cristalinas con sus energías electrónicas. De acuerdo con *DFT* o cualquier otro método de estructura electrónica, la energía total depende de las posiciones de los núcleos atómicos (*cf.* sección 2.2); por lo tanto, una métrica en el espacio configuracional adecuada debería ser capaz de reproducir hasta cierto punto esta relación configuración-energía, lo cual no obstante no se logra con ninguna de las métricas propuestas. Esto muestra que las descripciones de las estructuras cristalinas a través de estas métricas destacan aspectos diferentes que podrían ser útiles al momento de distinguir entre polimorfos, cuyas energías son muy similares. Esto último considerando también que, como se mostró previamente, las medidas detectan diferencias entre polimorfos (figs. 19 y 20). Así, se tienen indicios de que las métricas propuestas pueden ser útiles en la predicción de estructuras cristalinas; sin embargo, dada la variabilidad de estas métricas no existe manera de asegurar que estas lleven a una predicción de la estructura cristalina de los compuestos analizados adecuada.

l	Clúster	$E_{min} / kcalmol^{-1}$	$\bar{E} / kcalmol^{-1}$	$\sigma(E)$	PR (%)	CV(RMSD)
14	1	-3,160.13	-3116.84	15.66	71	1.04
	2	-3,155.76	-3108.11	17.68	29	1.05
26	1	-3,160.13	-3114.74	16.59	96	1.04
	2	-3,142.79	-3104.52	17.15	4	1.25
107	1	-3,160.13	-3115.43	16.72	81	1.08
	2	-3,148.44	-3109.68	15.94	19	1.20

Tabla 6: Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con las matrices de distancia $\mathbf{D}_{bmm}^z$ que fueron calculadas con 14, 26 y 107 moléculas cercanas (l) en relación a la molécula de referencia. Para los clústeres obtenidos se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los *medoides* ($CV(RMSD)$).

Además, los coeficientes de variación de las distancias de las estructuras cristalinas a sus *medoides*, CV , en cada clúster son en la mayoría de los casos menores para las matrices de z (tabs. 5 y 6) que para $\mathbf{D}_{8,bmp}^{RCD}$ (tab. 7). Por lo tanto, las medidas $\mathbf{D}^z$ generan clústeres con una menor variabilidad en las configuraciones de sus es-

l	Clúster	$E_{min} / kcalmol^{-1}$	$\bar{E} / kcalmol^{-1}$	$\sigma(E)$	PR (%)	CV(RMSD)
8	1	-3,160.13	-3110.78	16.64	47	1.36
	2	-3,156.79	-3114.22	16.33	36	1.40
	3	-3,158.72	-3124.59	13.25	17	1.42

Tabla 7: Análisis de los clústeres obtenidos al analizar 15594 estructuras cristalinas de la formamida con la matriz de distancia $\mathbf{D}_{8,bmp}^{RCD}$. Para cada clúster se reportan el valor más bajo de energía (E_{min}), el promedio ($\bar{E}$) y la desviación estándar ($\sigma(E)$) de las energías electrónicas, la población relativa (PR) y el coeficiente de variación de las distancias respecto a los *medoides* ($CV(RMSD)$).

estructuras cristalinas. Aun así, los valores del CV muestran para todas las matrices de distancia valores mayores a uno, lo cual implica que el $RMSD$ de las distancias a los *medoides* es mayor al promedio de estas. Así, existe una gran variación de las configuraciones de las estructuras cristalinas en todos los clústeres generados. Esto implica que los clústeres óptimos (máximo valor del coeficiente de silueta promedio) obtenidos con $\mathbf{D}_{107,bmp}^z$ y $\mathbf{D}_{107,bmm}^z$ (figs. 21 y 23) representan aún conjuntos de estructuras vastos en términos de las configuraciones de las estructuras que los conforman. En consecuencia, aunque estas métricas ($\mathbf{D}_{bmp}^z$ y $\mathbf{D}_{bmm}^z$) son útiles en el agrupamiento de estructuras cristalinas, la comparación de los ambientes químicos alrededor de la molécula de referencia que estas generan no es suficiente para definir una separación de las estructuras cristalinas más precisa.

Por otra parte, se analizaron los valores del puntaje de silueta de las 15594 estructuras cristalinas de la formamida (fig. 25, pág. 133). Las gráficas mostradas en la figura 25 resultan del agrupamiento generado con diferentes matrices de distancia de las métricas z , RCD y ΔE . En cada gráfica los coeficientes de silueta están ordenados de tal forma que los valores correspondientes a las estructuras que constituyen un mismo clúster son contiguos, de manera análoga a una gráfica de barras con los valores del coeficiente de silueta en el eje x y la etiqueta del clúster en el eje y , tal que el ancho de cada conjunto de datos representa el tamaño de cada clúster. Este tipo de gráficas fueron propuestas por Rousseeuw (1987) y han sido ampliamente usadas para la validación e interpretación en general de métricas empleadas en métodos de agrupamiento (Bora y Gupta 2014).

En el caso de la medida $\mathbf{D}^z$ (figs. 25a a 25f), conforme aumenta l incrementa la cantidad de coeficientes de silueta negativos; es decir, el número de estructuras cristalinas que no “encajan” en ninguno de los clústeres. Aumentar el valor de l tiene como efecto que la mayoría de las estructuras cristalinas se acomoda en un único clúster, cuyos coeficientes de silueta se encuentran por encima de la media. Por lo tanto, el puntaje de silueta promedio para las matrices $\mathbf{D}_{107,bmp}^z$, $\mathbf{D}_{26,bmm}^z$ y $\mathbf{D}_{107,bmm}^z$

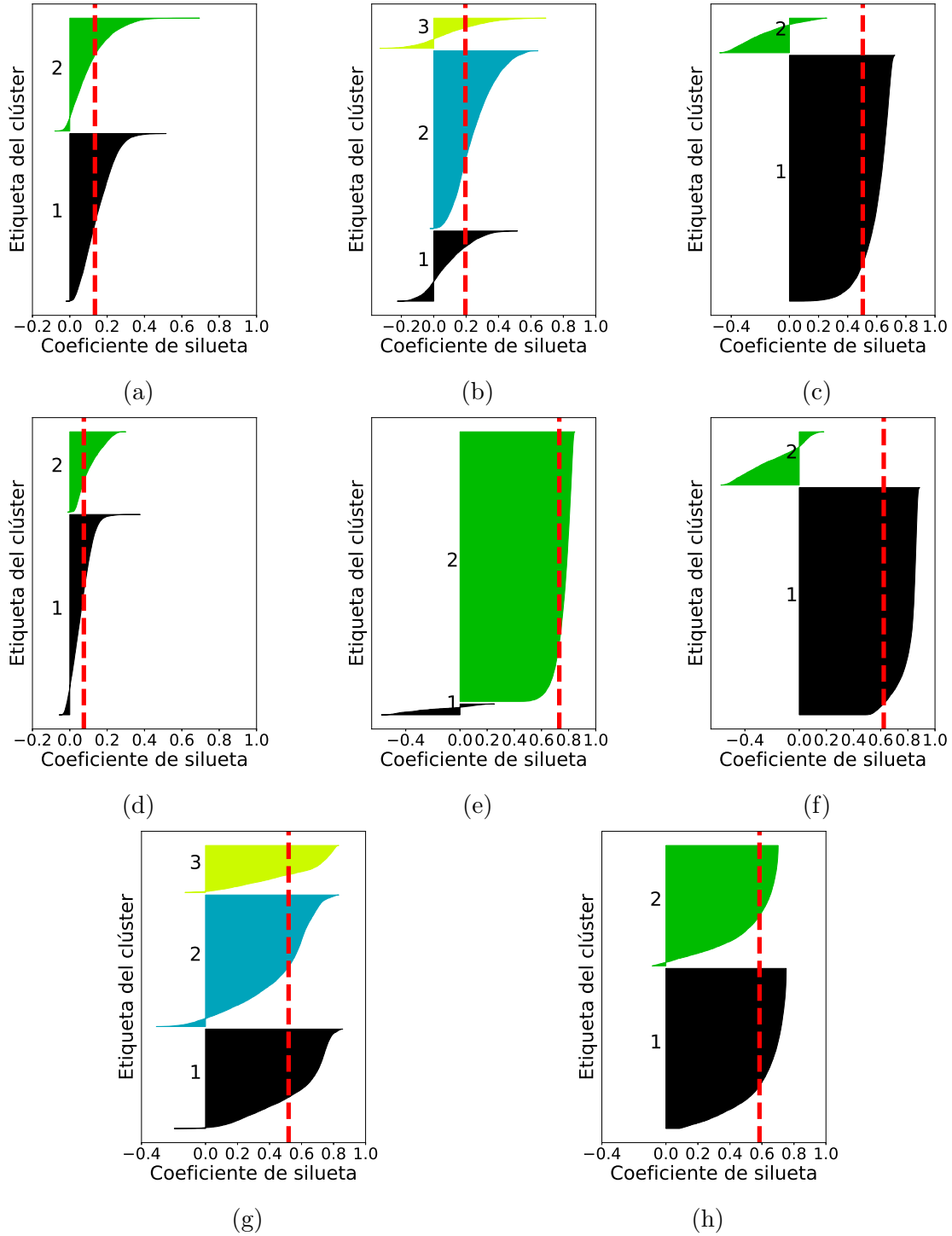


Figura 25: Coeficiente de silueta para cada estructura cristalina de la formamida en los clústeres obtenidos con las matrices de distancia (a) $D_{14,bmp}^z$, (b) $D_{26,bmp}^z$, (c) $D_{107,bmp}^z$, (d) $D_{14,bmm}^z$, (e) $D_{26,bmm}^z$, (f) $D_{107,bmm}^z$, (g) D_8^{RCD} y (h) $D^{\Delta E}$. Así, las primeras dos filas corresponden a la medida z , la primera con el algoritmo bmp y la segunda con bmm , los valores de l aumentan de izquierda a derecha. El coeficiente de silueta promedio se muestra con la línea roja discontinua.

(figs. 25c, 25e y 25f) queda determinado por los valores individuales del coeficiente de silueta de las estructuras asociadas a un único clúster. Así, la métrica z lleva a

una división binaria del espacio configuracional: estructuras cristalinas con configuraciones similares y aquellas con configuraciones distintas. Así, aunque la métrica z es capaz de distinguir en cierta medida entre las configuraciones de las estructuras, la descripción del espacio configuracional a través de las matrices de distancia es insuficiente para generar una mayor subdivisión del conjunto completo de estructuras (mayor cantidad de clústeres). Esto coincide con lo discutido en la sección previa: al aumentar el valor de l se pierde información de la comparación de moléculas por pares, tal que todas las estructuras cristalinas terminan siendo similares. En otras palabras, se pierde cierto detalle en la descripción del espacio configuracional. Esto a su vez coincide con que en el clúster más poblado todos los valores del coeficiente de silueta son cercanos. No obstante, esto podría implicar que existe una sola estructura representativa (clúster más poblado). De acuerdo con lo reportado en la literatura, hasta el momento se conoce experimentalmente una única estructura cristalina de la formamida en condiciones estándares de temperatura y presión (Gajda y Katrusiak 2011). Por lo tanto, podría pensarse que se debería encontrar un único clúster al realizar el agrupamiento del conjunto de estructuras cristalinas que se generaron aleatoriamente. Sin embargo, como se verá más adelante, no existe una verdadera correlación entre el número de estructuras cristalinas caracterizadas experimentalmente y el número de clústeres, al menos al nivel del agrupamiento generado con las métricas empleadas en esta tesis.

Por el contrario, la medida RCD resulta en una distribución de los coeficientes de silueta similar entre los tres clústeres que genera (fig. 25g). En este caso, la mayoría de los valores del puntaje de silueta se encuentran por encima del promedio calculado para estas tres agrupaciones. Así, el coeficiente de silueta promedio es representativo de todos los clústeres, a diferencia de los resultados obtenidos con las matrices $\mathbf{D}^z$; tal que la medida $\mathbf{D}_{8,bmp}^{RCD}$ estaría indicando la potencial existencia de tres estructuras cristalinas para la formamida. Aun así, para cada clúster se observan valores del coeficiente de silueta negativos que, como se mencionó anteriormente, indican una asignación incorrecta y la existencia de estructuras cristalinas aisladas o con configuraciones demasiado diferentes. Sin embargo, esto está en acuerdo con gráficas de los valores individuales del coeficiente de silueta de este tipo que se han reportado para otro tipo de métricas empleadas con métodos de agrupamiento (Bora y Gupta 2014, Rousseeuw 1987) y con la gráfica obtenida empleando las diferencias de energía ($\mathbf{D}^{\Delta E}$, fig. 25h). Por lo tanto, el RCD permite generar una mayor partición del espacio configuracional, tal que sería factible generar un mayor número de clústeres del conjunto de estructuras cristalinas de la formamida, aunque estos no generasen el valor máximo del coeficiente de silueta promedio, tal como

proponen Li et al. (2018). Pero esto difiere del objetivo de generar los clústeres sin imponer arbitrariamente el número de estos. Además, los clústeres generados con $\mathbf{D}_{bmp}^{RCD}$, como se discutió previamente, presentan una alta variabilidad en las configuraciones de las estructuras cristalinas que los constituyen (*CV*, tab. 7). Por lo tanto, es importante remarcar que aunque la medida $\mathbf{D}_{8,bmp}^{RCD}$ genera una separación de las estructuras cristalinas más homogénea (fig. 25g), cada clúster representa un amplio espectro de posibles configuraciones, en términos del *CV(RMSD)*.

De acuerdo con lo discutido en esta sección, las métricas en el espacio configuracional propuestas permiten agrupar las estructuras cristalinas de la formamida. Sin embargo, los coeficientes de variación (*CV*) de las distancias a los *medoides* en los clústeres fueron mayores a uno; por lo tanto, las agrupaciones generadas representan un espectro amplio en términos de las configuraciones de las estructuras que constituyen cada clúster, según cada métrica. Esto puede entenderse como que cada clúster representa una región amplia en una superficie de energía potencial, la cual podría delimitar los puntos de mínima energía. Sin embargo, la cantidad de estructuras cristalinas que deberían optimizarse para identificar las de mínima energía es aún demasiado vasta. Por lo tanto, es necesario mejorar las métricas con el fin de generar clústeres de estructuras cristalinas menos dispersos (en términos de las configuraciones de las estructuras que los constituyen) con el fin de delimitar mejor la información relevante, lo cual se discutirá en la sección de perspectivas. Además, en el caso de las medidas $\mathbf{D}^z$ los resultados obtenidos varían de manera considerable según el número de moléculas vecinas en relación a la molécula de referencia considerado (*l*). En consecuencia, las diferentes maneras de calcular este tipo de matrices de distancia podrían llevar a análisis muy distintos. Entonces, con el fin de corroborar lo presentado en esta sección y descartar que la identificación con las métricas $\mathbf{D}_{107,bmp}^z$ y $\mathbf{D}_{107,bmm}^z$ de un solo clúster conformado por la mayoría de las estructuras cristalinas de la formamida (figs. 25c y 25f) se deba al compuesto químico analizado, se analizarán estructuras cristalinas del ácido oxálico y de la hidrazida maleica.

6.5. Análisis de sistemas polimorficos

De acuerdo con los análisis presentados en la sección anterior, las matrices de distancia $\mathbf{D}^z$ son adecuadas para generar clústeres a partir del conjunto de estructuras cristalinas de la formamida con valores del coeficiente de silueta altos. Sin embargo, las configuraciones de las estructuras en cada clúster son aún muy variadas. Por lo tanto, se consideraron también los compuestos hidrazida maleica y ácido oxálico, cuyos cristales presentan polimorfismo, con el fin de corroborar si los clústeres determinados en estos casos podrían representar redes cristalinas distintas. Se realizó un análisis para las estructuras cristalinas generadas aleatoriamente para estos compuestos similar al presentado para la formamida en la sección previa. Se analizaron 1092 estructuras cristalinas de la hidrazida maleica y 970 estructuras del ácido oxálico, de acuerdo con lo descrito en la metodología (*cf.* sección 5.1).

En el caso de las estructuras cristalinas de la hidrazida maleica, al variar el parámetro l en el cálculo de las matrices de distancia no se observó un cambio importante en las tendencias del coeficiente de silueta promedio (fig. 26), en relación a las observadas en el análisis de las estructuras de la formamida (fig. 21). En el caso de las métricas $\mathbf{D}^z$ y $\mathbf{D}^{RMSD}$ calculadas con los algoritmos de búsqueda de mínimos *bmp* y *bmm*, el puntaje de silueta promedio aumenta conforme incrementa l ; mientras que para la medida $\mathbf{D}_{bmm}^{RCD}$ este decrece. La métrica D_{bmm}^z genera mayores fluctuaciones del coeficiente de silueta promedio al modificar l , lo que se debe a la propia metodología. Aun así, el algoritmo *bmm* también genera en este caso los valores más altos del coeficiente de silueta promedio.

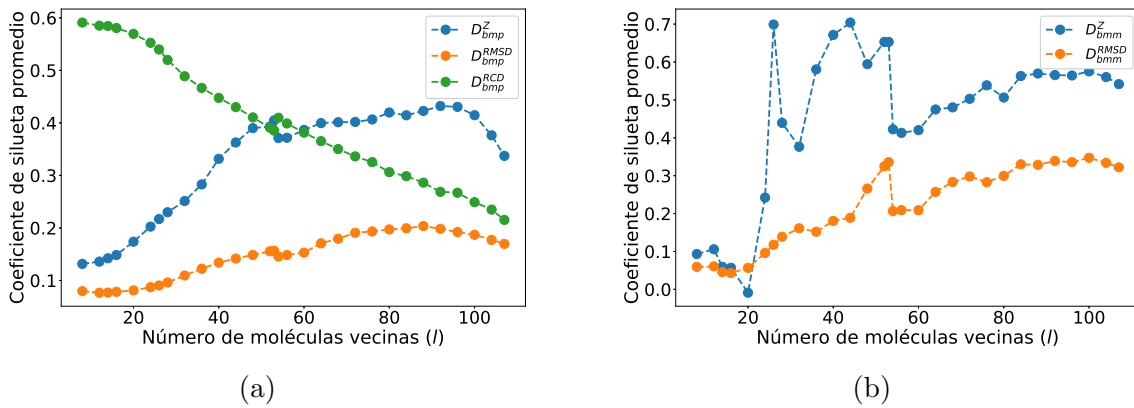


Figura 26: Coeficiente de silueta promedio de los clústeres obtenidos con las métricas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (MH); en todos los casos se usó el método de agrupamiento espectral.

Por el contrario, en el caso de las estructuras cristalinas del ácido oxálico, el

valor máximo del coeficiente de silueta promedio es más pronunciado para la matriz D_{bmp}^z (27a); es decir, no se observan valores cercanos al valor máximo del coeficiente de silueta promedio (en $l = 96$) como en los casos anteriores (figs. 21a y 26a). Así, para las estructuras cristalinas del ácido oxálico se tiene una mayor variación en la tendencia de los valores del puntaje de silueta promedio. El cambio en el coeficiente de silueta promedio que resulta al variar de 53 a 54 moléculas vecinas (l), valores para los cuales se modifican las matrices de comparación, es mayor al observado para *MH*. Por lo tanto, la cantidad de moléculas vecinas (l) que lleva a las mejores agrupaciones se modifica de manera importante de un compuesto a otro, tal que sería necesario realizar un análisis como el que se presenta aquí para cada compuesto que se requiera estudiar. No obstante, aunque en este caso el uso de $l = 107$ para los siguientes análisis representa un cambio importante respecto a las agrupaciones óptimas (coeficiente de silueta máximo), este valor de l se seguirá empleando con el fin de permitir de forma directa la comparación con los sistemas previamente analizados. Además, tanto para el valor de l que corresponde al máximo del coeficiente de silueta promedio ($l = 96$) como para $l = 107$, se obtuvieron 2 clústeres (Anexo A, fig. 36a), lo cual permite mantener en cierta medida un caso similar al óptimo (puntaje de silueta promedio máximo).

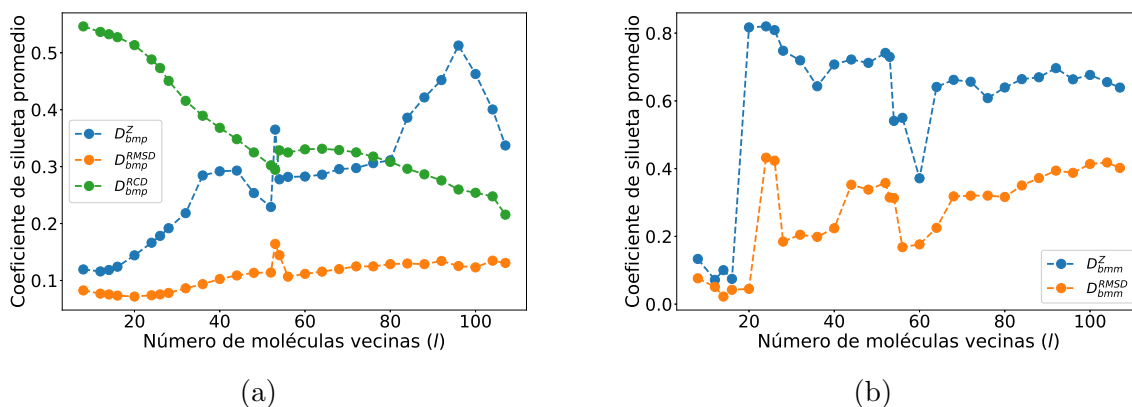


Figura 27: Coeficiente de silueta promedio de los clústeres obtenidos con las métricas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 970 estructuras cristalinas del ácido oxálico (OA); en todos los casos se usó el método de agrupamiento espectral.

En la figura 28 se muestran los valores del coeficiente de silueta para las estructuras cristalinas de la hidrazida maleica generadas aleatoriamente, que fueron agrupadas con las matrices de distancia $D_{107,bmp}^z$, $D_{107,bmm}^z$ y D_{bmp}^{RCD} . El número de clústeres generados con $D_{107,bmm}^z$ es igual al número de polimorfos conocidos para *MH* (tres); mientras que la medida $D_{107,bmp}^z$ sigue generando solo dos clústeres. Por lo tanto, el algoritmo *bmm* permite generar en este caso una mayor subdivisión de

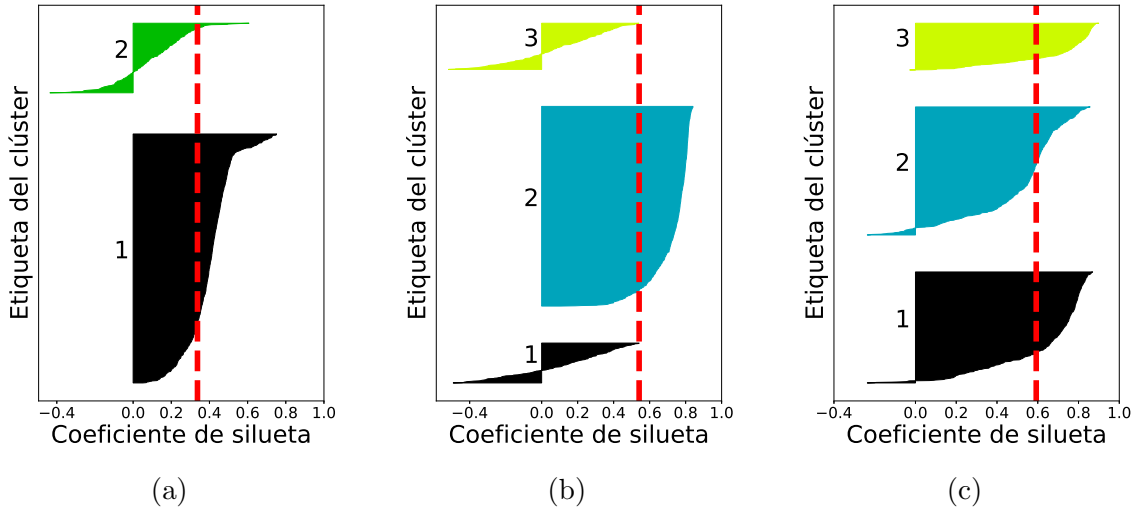


Figura 28: Coeficientes de silueta para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (*MH*) de acuerdo con los clústeres obtenidos con las matrices de distancia (a) $D_{107,bmp}^z$, (b) $D_{107,bmm}^z$ y (c) D_8^{RCD} . El coeficiente de silueta promedio se muestra con la línea roja discontinua.

las estructuras cristalinas, en comparación con los clústeres obtenidos para las estructuras cristalinas de la formamida. Sin embargo, las distribuciones de los valores del puntaje de silueta generadas con $\mathbf{D}_{107,bmm}^z$ y $\mathbf{D}_{107,bmp}^z$ son similares a las que se generaron para la formamida (fig. 25). Es decir, solo uno de los clústeres da como resultado la mayoría de los valores del coeficiente de silueta por encima de la media. Así, las estructuras en el resto de los clústeres obtenidos con $\mathbf{D}^z$ (considerando *bmp* y *bmm*, figs. 28a y 28b) corresponden a valores del coeficiente de silueta menores al promedio e incluso se tienen valores negativos, lo cual muestra una asignación que no es óptima en comparación con los clústeres más poblados (clústeres 1 y 2 en las figuras 28a y 28b, respectivamente).

Además, aunque en el caso de la matriz $\mathbf{D}_{107,bmm}^z$ el número de clústeres óptimo es igual al número de polimorfos de la hidrazida maleica, los clústeres no representan realmente tres estructuras cristalinas potenciales. Considerando estos clústeres, se identificó que las estructuras cristalinas más similares (menor distancia $\mathbf{D}_{107,bmm}^z$) a los tres polimorfos determinados experimentalmente se encuentran en un único clúster, el más poblado (clúster 2, fig. 28b). Esto corrobora que la comparación de las configuraciones de las estructuras cristalinas generada con las matrices de distancia $\mathbf{D}^z$ carece de detalle, tal que no es factible subdividir el clúster de estructuras más poblado e identificar un clúster por cada estructura cristalina conocida experimentalmente.

En el caso de la matriz de distancia $\mathbf{D}_{8,bmp}^{RCD}$, las estructuras cristalinas semejantes a las estructuras de los polimorfos *MH1* y *MH3* se encuentran en el clúster 3,

mientras que la estructura semejante a la de *MH2* se encuentra en el clúster 2 (fig. 28c). Así, la métrica $\mathbf{D}_{8,bmp}^{RCD}$ permite una mejor separación de las estructuras cristalinas de la hidrazida maleica; sin embargo, tampoco se obtiene un clúster por cada estructura identificada experimentalmente. Además, de acuerdo con los valores de distancia entre las estructuras caracterizadas experimentalmente obtenidos con la misma métrica ($\mathbf{D}_{8,bmp}^{RCD}$), existe una mayor similitud entre las configuraciones de los polimorfos *MH1* y *MH2* ($d_{8,bmp}^{RCD} = 0.21$) que entre las de *MH1* y *MH3* ($d_{8,bmp}^{RCD} = 0.35$). Por lo que el hecho de que las estructuras semejantes a *MH1* y *MH2* se hayan identificado en clústeres diferentes también implica que las agrupaciones no son óptimas.

En el caso del ácido oxálico, las gráficas del coeficiente de silueta para las 970 estructuras generadas aleatoriamente son similares a las presentadas para la formamida (fig. 25). Las estructuras cristalinas generadas aleatoriamente e identificadas como semejantes a las estructuras caracterizadas experimentalmente se encontraron en un único clúster con todas las matrices de distancia ($\mathbf{D}_{bmp}^{RCD}$, $\mathbf{D}_{107,bmp}^z$, $\mathbf{D}_{107,bmm}^z$). Por la similitud en el resultado del análisis de este conjunto de estructuras cristalinas, las gráficas correspondientes a *OA* se incluyeron en el anexo A (fig. 37, pág. 165).

De forma análoga al caso de la formamida, para la hidrazida maleica y el ácido oxálico se calcularon también las distancias ($\mathbf{D}^z$, $\mathbf{D}^{RMSD}$ y $\mathbf{D}_{bmp}^{RCD}$) entre las estructuras cristalinas generadas de manera aleatoria y las estudiadas experimentalmente. Así, para cada polimorfo (tres de *MH* y dos de *OA*) se identificó la estructura generada con *Genarris* que le fuese configuracionalmente cercana, según las métricas. En la figura 29 se muestran las diferencias promedio entre los vectores que definen a la celda unitaria ($\vec{a}$, $\vec{b}$, y $\vec{c}$) experimentales y de las estructuras cristalinas generadas más semejantes (ec. 113). En este análisis se consideraron valores de $l=8, 14, 26$ y 107 con el fin de tener una mayor cantidad de datos, entre los cuales comparar las diferentes métricas.

Como se pudo observar en el análisis de las métricas (cf. sección 6.3) y de las estructuras cristalinas de la formamida, la modificación del número de moléculas vecinas (l) al calcular las matrices de distancia genera comparaciones distintas de las configuraciones de las estructuras cristalinas del conjunto analizado. En la figura 29 se puede observar que, por ejemplo, al emplear la métrica $\mathbf{D}_{bmp}^{RCD}$ las diferencias entre los vectores que definen la celda unitaria de las estructuras experimentales y de las estructuras generadas aleatoriamente e identificadas como semejantes son menores que al emplear cualquier otra de las métricas ($\mathbf{D}^z$ y $\mathbf{D}^{RMSD}$, calculadas con el algoritmo *bmp* o *bmm*). Incluso, en el caso de las estructuras del ácido oxálico

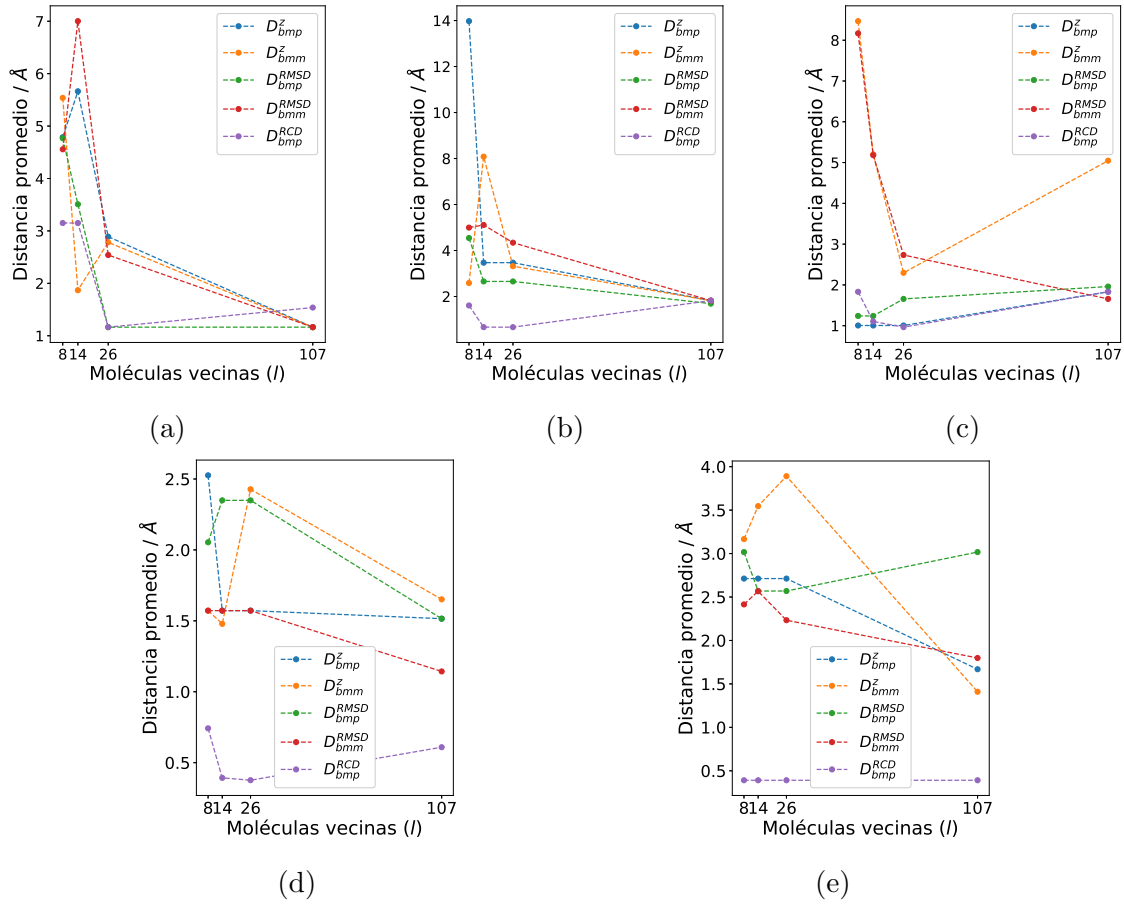


Figura 29: Distancia promedio entre los vectores que definen la celda unitaria de las estructuras determinadas experimentalmente (a) *MH1*, (b) *MH2*, (c) *MH3*, (d) *OA* – α y (e) *OA* – β y de las estructuras cristalinas generadas aleatoriamente que fueron identificadas como similares con las métricas z , $RMSD$ y RCD . En el caso de las medidas z y $RMSD$ se emplearon los algoritmos *bmp* y *bmm*. Todas las matrices de distancia se calcularon considerando $l=8, 14, 26$ y 107 .

(figs. 29d y 29e), estas diferencias llegan a ser inferiores a $0.5 \text{ \AA}$. Sin embargo, para la misma métrica, la diferencia llega a ser superior a $3.0 \text{ \AA}$ para el polimorfo *MH1* al considerar el parámetro l igual a 8 (fig. 29a), el cual fue identificado como óptimo para esta métrica (fig. 26a). En el caso de *MH2* (fig. 29b) y *MH3* (fig. 29c) los mejores resultados (menor diferencia entre los vectores que definen la celda unitaria) se obtienen para $l = 14$ y $l = 26$, mientras que los peores resultados se obtienen con $l = 8$. Así, la métrica RCD funciona de manera adecuada para todos los polimorfos principalmente para los valores de $l = 14$ y $l = 26$, pero existen excepciones y la misma medida puede dar como resultado la correlación de las estructuras cristalinas experimentales con estructuras generadas muy diferentes, en términos de sus celdas unitarias. Estos cambios en el comportamiento de $\mathbf{D}_{bmp}^{RCD}$ según el compuesto analizado o el número de vecinos cercanos (l) ocurre para todas las métricas.

Al considerar 107 moléculas vecinas en el cálculo de las matrices de distancia

$\mathbf{D}^z$ y $\mathbf{D}^{RMSD}$ se generan las mejores agrupaciones (considerando ambos algoritmos de búsqueda de mínimos), pero no fue posible identificar en todos los casos las estructuras cristalinas con las celdas unitarias más similares a las determinadas experimentalmente. Para estas métricas la diferencia entre los vectores que definen a la celda unitaria de las estructuras experimentales y de las generadas aleatoriamente suele disminuir conforme aumenta el número de moléculas vecinas (fig. 29); por ejemplo, en el caso de los polimorfos *MH1* (fig. 29a) y *MH2* (fig. 29b) las medidas $\mathbf{D}_{107}^z$ y $\mathbf{D}_{107}^{RMSD}$ (con *bmp* y *bmm*) son comparables a $\mathbf{D}_{bmp}^{RCD}$. Sin embargo, en el caso de *MH3* (fig. 29c) los mejores resultados se obtienen con las matrices $\mathbf{D}_{8,bmp}^z$, $\mathbf{D}_{8,bmp}^{RMSD}$, $\mathbf{D}_{26,bmm}^z$ y $\mathbf{D}_{107,bmm}^{RMSD}$. Por lo tanto, ningún valor de l asegura un resultado confiable en todas las comparaciones realizadas con las matrices $\mathbf{D}^z$ y $\mathbf{D}^{RMSD}$. Esta misma conclusión puede obtenerse al analizar la diferencia relativa entre el volumen de la celda unitaria determinada experimentalmente y de la estructura generada más semejante, lo cual se incluye en el anexo A (fig. 38, pág. 165).

De acuerdo con el análisis realizado en esta sección, las métricas propuestas son adecuadas para agrupar las estructuras cristalinas generadas aleatoriamente para los compuestos analizados en esta tesis (formamida, hidrazida maleica y ácido oxálico), pero los clústeres generados no representan necesariamente estructuras potenciales diferentes. Además, las comparaciones de las configuraciones de estructuras cristalinas dadas por estas métricas cambian de manera importante de un compuesto a otro y en dependencia del número de moléculas cercanas en relación a la molécula de referencia considerado en el cálculo de las matrices de distancia. Por lo tanto, los clústeres de estructuras con configuraciones moleculares similares generados con una misma métrica (z , RCD o $RMSD$), pero considerando valores distintos de l en el cálculo de las matrices de distancia, pueden ser muy diferentes y dar como resultado la predicción de estructuras cristalinas a partir de estos clústeres muy distintas.

7. Conclusiones

En este trabajo se desarrolló una nueva metodología para la descripción y la comparación de la posición del centro geométrico y la orientación en relación a una molécula de referencia (configuración) de moléculas dentro de redes cristalinas. En particular, se planteó describir la configuración de las moléculas en las redes cristalinas con base en un conjunto de cuaterniones duales. Con el fin de comparar un gran número de estructuras cristalinas que fueron generadas aleatoriamente con el programa *Genarris* para los compuestos formamida, hidrazida maleica y ácido oxálico, se propusieron tres métricas o medidas de disimilitud entre pares de cuaterniones duales (s , z y z_{rc}) que permitieron establecer matrices de comparación, **C**. Cada matriz de comparación, **C**, constituye el cálculo de alguna de las métricas entre todos los pares de moléculas de un par de estructuras. Las matrices **C** fueron reducidas a valores de distancia entre las configuraciones de pares de estructuras a través de la búsqueda de los valores más pequeños en estas matrices con dos algoritmos diferentes: *bmp* y *bmm*. Este procedimiento permitió obtener métricas que reflejan la diferencia entre las configuraciones de las moléculas en las estructuras cristalinas. Los valores de las distancias entre todos los pares de estructuras cristalinas de un mismo compuesto molecular se almacenaron en matrices de distancia, **D**, las cuales a su vez se emplearon para generar clústeres con métodos de agrupamiento. Los clústeres son conjuntos de estructuras cristalinas que han sido identificadas como similares en términos de la configuración de sus moléculas de acuerdo con cada una de las métricas empleadas, a partir de la subdivisión de un conjunto de estructuras cristalinas generadas aleatoriamente.

En la primera etapa del proyecto se mostró que los programas desarrollados en esta tesis funcionan de manera adecuada. Para la comparación de un par de estructuras cristalinas se propuso un cambio de base o de marco de referencia de los cuaterniones duales que definen una de las estructuras en relación a la orientación de la molécula de referencia de la segunda estructura (ec. 101). Tal que la comparación de los conjuntos de cuaterniones duales de dos estructuras cristalinas fuese independiente de la molécula de referencia elegida en cada sistema. Se demostró numéricamente que las métricas propuestas son invariantes ante la rotación de las estructuras cristalinas comparadas, tal que se corroboró que los cambios de base con cuaterniones duales propuestos son correctos. Por otra parte, se probaron diferentes métodos de agrupamiento para generar clústeres de estructuras cristalinas con configuraciones similares, entre los cuales el método *espectral* generó los mejores resultados.

En la segunda parte, se mostró que las métricas z y z_{rc} , que reflejan la diferencia entre la orientación relativa de pares de moléculas en redes cristalinas, generan una descripción del ambiente químico alrededor de una molécula de referencia similar al MSD y al $RMSD$, respectivamente. Por el contrario, se encontró que la medida s no genera una buena descripción, ya que detecta muy pocos cambios en las orientaciones de las moléculas, en comparación con el $RMSD$. Además, se mostró que los algoritmos de búsqueda de mínimos bmp y bmm pueden dar como resultado la pérdida de información importante de la comparación por pares de moléculas al reducir las matrices de comparación, $\mathbf{C}$, (comparación por pares de moléculas) y calcular la distancia entre pares de estructuras cristalinas (cálculo de la matriz $\mathbf{D}$). Se mostró también que las matrices $\mathbf{D}^z$ permiten distinguir entre configuraciones de polimorfos de un mismo compuesto molecular, al igual que las matrices $\mathbf{D}^{RMSD}$ y $\mathbf{D}_{bmp}^{RCD}$. Sin embargo, al calcular las matrices de distancia considerando diferentes valores de moléculas vecinas (l) en relación a la molécula de referencia, se generaron comparaciones muy diferentes de las configuraciones de las estructuras cristalinas. Por ejemplo, la métrica $\mathbf{D}_{bmp}^z$ para $l \leq 100$ identifica que los polimorfos $MH2$ y $MH3$ son similares entre sí y diferentes de $MH1$, pero para $l \geq 107$ la misma métrica indica lo contrario.

En la penúltima sección se estudió con las matrices de distancia $\mathbf{D}_{bmp}^z$, $\mathbf{D}_{bmm}^z$ y $\mathbf{D}_{bmp}^{RCD}$ un conjunto de 15594 estructuras cristalinas de la formamida que fueron generadas aleatoriamente con el programa *Genarris*. Para este conjunto de estructuras fue posible generar clústeres con el método de agrupamiento espectral con un valor alto del puntaje de silueta promedio con ambas matrices de distancia $\mathbf{D}^z$, que fueron similares a los obtenidos con la medida $\mathbf{D}_{bmp}^{RCD}$. Además, las tendencias de los valores del puntaje de silueta promedio fueron similares a las obtenidas con $\mathbf{D}^{\Delta E}$, la cual es la matriz de diferencia de energía electrónica entre todos los pares de estructuras analizadas y que se usó como punto de referencia. Comparando los algoritmos bmp y bmm , la medida $\mathbf{D}_{bmm}^z$ generó los mejores resultados. Sin embargo, la calidad del resultado dependió del número de moléculas vecinas que se consideraron en el ambiente químico de la molécula de referencia al calcular las matrices de distancia (en particular, $l = 8, 14, 26, 107$). Los clústeres generados representaron conjuntos vastos en términos de las configuraciones de las estructuras que los conformaron, tal que no fue posible asegurar que cada clúster representase un conjunto de estructuras relevante para la predicción de la estructura cristalina de la formamida.

Al analizar 1094 estructuras cristalinas de la hidrazida maleica y 970 del ácido oxálico, no se encontró con ninguna de las matrices de distancia una correlación entre el número de clústeres óptimos (correspondientes al mayor puntaje de silueta prome-

dio) y el número de estructuras cristalinas conocidas experimentalmente. Asimismo, se corroboró que los clústeres de estructuras cristalinas generados se modifican de manera importante al variar el valor del parámetro l en el cálculo de las matrices de distancia. Además, la correlación entre este parámetro y los clústeres generados también dependió del compuesto que conforma las estructuras cristalinas, tal que los valores de l que generan los mejores clústeres (valores máximos del puntaje de silueta) fueron diferentes para cada conjunto de estructuras cristalinas analizado. Por lo tanto, las matrices de distancia propuestas llevan a una gran cantidad de posibilidades de la comparación de las estructuras cristalinas, y por lo tanto, de los clústeres relevantes al agrupar los conjuntos de estructuras cristalinas generadas aleatoriamente, tal que no fue posible identificar la matriz de distancia que asegurase los mejores resultados para todos los conjuntos de estructuras analizados. La división del espacio configuracional generada con estas matrices carece de precisión, tal que se vuelve inviable partir de cada uno de los clústeres para la predicción de las estructuras cristalinas que pudiesen identificarse experimentalmente para cada compuesto.

En conclusión, los clústeres de estructuras generados con las matrices $\mathbf{D}^z$ y el método de agrupamiento espectral para cada conjunto de estructuras analizado no cumplen con el objetivo planteado en esta tesis y, por lo tanto, aún es necesario mejorar la comparación de la configuración de moléculas en redes cristalinas. No obstante, esto se debe principalmente a la elección del parámetro l para el cálculo de las matrices de distancia. Se reitera que la descripción de estructuras cristalinas moleculares con cuaterniones duales funciona y puede emplearse incluso con una perspectiva diferente a la de este trabajo.

8. Perspectivas

Con base en lo anterior, la métrica z permitió describir el ambiente químico alrededor de una molécula de referencia de manera similar a las métricas RCD y $RMSD$ (cf. sección 6.3). Sin embargo, no fue posible definir una subdivisión de las estructuras cristalinas de cada conjunto generado aleatoriamente suficientemente precisa a través de los clústeres generados con las matrices de distancia $\mathbf{D}^z$. Por lo tanto, será necesario extender el análisis de la métrica z entre cuaterniones duales con el fin de optimizarla. Por ejemplo, se podrían comparar las orientaciones relativas de solo un par de moléculas y rotar una de las moléculas gradualmente, tal que se pueda evaluar si existen orientaciones particulares para las cuales la métrica no es capaz de detectar similitudes o diferencias. De ser así puede que sea necesario proponer otra métrica con base en el producto de cuaterniones duales. Por otra parte, la transformación de las matrices de comparación, $\mathbf{C}$, a valores de distancia con los algoritmos *bmm* y *bmp* llevó a una gran cantidad de posibles matrices de distancia en términos del parámetro l . Esto fue uno de los principales inconvenientes al momento de realizar el análisis configuracional de las estructuras cristalinas. Por lo tanto, creemos que una manera de mejorar la metodología presentada en esta tesis es evitar la transformación de las matrices $\mathbf{C}$ a valores de distancia. Por ejemplo, previo a la comparación de todos los pares posibles de estructuras cristalinas, pueden compararse las orientaciones moleculares relativas (representadas con cuaterniones duales) en una misma estructura cristalina. Esto permitiría describir el ambiente químico alrededor de la molécula de referencia, lo cual funcionó adecuadamente, igual construyendo una matriz de comparación, pero que en este caso dependería de solo una estructura. Posteriormente, reducir esta matriz a un valor que sea representativo de la estructura cristalina. De esta manera, se obtendría un valor que describiría el ambiente químico en cada estructura cristalina, tal que la comparación entre pares de estructuras se haría a través de las diferencias absolutas de estos valores, de manera similar a lo propuesto por Sadeghi et al. (2013).

Asimismo, de acuerdo con lo planteado en la introducción, la metodología desarrollada en este trabajo mantiene un enfoque en la predicción de estructuras cristalinas. Por lo tanto, una vez que se hayan optimizado las métricas en el espacio configuracional, se plantea conjuntar esta metodología con el cálculo de la entropía configuracional de cristales moleculares. Esto considerando que sea factible establecer clústeres de estructuras cristalinas con configuraciones similares, tal que para cada uno de estos se puede identificar un sistema representativo y aquellos que le son similares podrían emplearse como un estimado de las posibles configuraciones

que el mismo sistema puede adoptar. Esto último pretende usarse en el cálculo de la entropía configuracional siguiendo modelos como el propuesto por Mayer y Wallek (2020). Así, se espera que al sumar la entropía configuracional a la energía interna de los sistemas, obtenida con métodos de estructura electrónica, se obtenga una mejor descripción de la energía libre de los mismos y, en consecuencia, se mejore la predicción de las estructuras cristalinas más estables.

Referencias

Adasme-Carreño, F., Caballero, J. y Ireta, J. (2021), ‘Psique: Protein secondary structure identification on the basis of quaternions and electronic structure calculations’, *Journal of Chemical Information and Modeling* **61**(4), 1789–1800. PMID: 33769809.

URL: <https://doi.org/10.1021/acs.jcim.0c01343>

AlQuraishi, M. (2019), ‘End-to-end differentiable learning of protein structure’, *Cell Systems* **8**(4), 292–301.e3.

URL: <https://www.sciencedirect.com/science/article/pii/S2405471219300766>

Ankerst, M., Breunig, M. M., Kriegel, H.-P. y Sander, J. (1999), Optics: Ordering points to identify the clustering structure, *in* ‘Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data’, SIGMOD ’99, Association for Computing Machinery, New York, NY, USA, p. 49–60.

URL: <https://doi.org/10.1145/304182.304187>

Aroyo, M. I., Perez-Mato, J. M., Capillas, C., Kroumova, E., Ivantchev, S., Mada-riaga, G., Kirov, A. y Wondratschek, H. (2006), ‘Bilbao crystallographic server: I. databases and crystallographic computing programs’, *Zeitschrift für Kristallographie - Crystalline Materials* **221**, 15 – 27.

Arthur, D. y Vassilvitskii, S. (2007), K-means++: The advantages of careful seeding, *in* ‘Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms’, SODA ’07, Society for Industrial and Applied Mathematics, USA, p. 1027–1035.

Bartók, A. P., Kondor, R. y Csányi, G. (2013), ‘On representing chemical environments’, *Phys. Rev. B* **87**, 184115.

URL: <https://link.aps.org/doi/10.1103/PhysRevB.87.184115>

Bell, E. W. y Zhang, Y. (2019), ‘Dockrmsd: an open-source tool for atom mapping and rmsd calculation of symmetric molecules through graph isomorphism.’, *Jour-*

- nal of cheminformatics* **11**(40), 1758–2946. PMID: 11456823.
URL: <https://doi.org/10.1186/s13321-019-0362-7>
- Bernstein, J. (2002), *Polymorphism in Molecular Crystals*, 1a edn, Oxford University Press, Oxford.
- Bhadeshia, H. K. D. H. (s.f.), ‘Lecture 7: Quasichemical solution models’.
URL: <https://www.phase-trans.msm.cam.ac.uk/mphil/MP4-7.pdf>
- Bhattacharya, S., Saraswatula, V. G. y Saha, B. K. (2013), ‘Thermal expansion in alkane diacids—another property showing alternation in an odd–even series’, *Crystal Growth & Design* **13**(8), 3651–3656.
URL: <https://doi.org/10.1021/cg400668w>
- Blöchl, P. E. (1994), ‘Projector augmented-wave method’, *Phys. Rev. B* **50**, 17953–17979.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.50.17953>
- Bonald, T. (2019-2020), ‘Spectral graph embedding’.
URL: <https://perso.telecom-paristech.fr/bonald/documents/spectral.pdf>
- Bora, D. J. y Gupta, A. K. (2014), ‘Effect of different distance measures on the performance of k-means algorithm: An experimental study in matlab’, *CoRR* **abs/1405.7471**.
URL: <http://arxiv.org/abs/1405.7471>
- Bottema, O. y Roth, B. (1979), *Theoretical kinematics*, 1a edn, General Publishing Company, Ontario.
- Brandenburg, J. G. y Grimme, S. (2016), ‘Organic crystal polymorphism: a benchmark for dispersion-corrected mean-field electronic structure methods’, *Acta Crystallographica Section B* **72**(4), 502–513.
URL: <https://doi.org/10.1107/S2052520616007885>
- Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E. y Dougherty, E. R. (2007), ‘Model-based evaluation of clustering validation measures’, *Pattern Recognition* **40**(3), 807–824.
URL: <https://www.sciencedirect.com/science/article/pii/S0031320306003104>
- Brunger, A. (1997), ‘X-ray crystallography and nmr reveal complementary views of structure and dynamics’, *Nature Structural Biology* **4**(S), 862–865.

- Bučko, T. c. v., Lebègue, S., Hafner, J. y Ángyán, J. G. (2013), ‘Tkatchenko-scheffler van der waals correction method with and without self-consistent screening applied to solids’, *Phys. Rev. B* **87**, 064110.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.87.064110>
- Celebi, M. E., Kingravi, H. A. y Vela, P. A. (2013), ‘A comparative study of efficient initialization methods for the k-means clustering algorithm’, *Expert Systems with Applications* **40**(1), 200–210.
URL: <https://www.sciencedirect.com/science/article/pii/S0957417412008767>
- Chang, G. (2015), ‘On least-squares solution to 3d similarity transformation problem under gauss-helmert model’, *Journal of Geodesy* **89**, 573–576.
- Chung, F. R. K. (1997), *Spectral Graph Theory*, Vol. 92, CBMS Regional Conference Series in Mathematics.
- Clifford (1871), ‘Preliminary sketch of biquaternions’, *Proceedings of the London Mathematical Society* **s1-4**(1), 381–395.
URL: <https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s1-4.1.381>
- Coutsias, E. A., Seok, C. y Dill, K. A. (2004), ‘Using quaternions to calculate rmsd’, *Journal of Computational Chemistry* **25**(15), 1849–1857.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20110>
- Cradwick, P. D. (1976), ‘Crystal structure of the growth inhibitor, ‘maleic hydrazide’(1,2-dihydropyridazine-3,6-dione)’, *J. Chem. Soc., Perkin Trans. 2* pp. 1386–1389.
URL: <http://dx.doi.org/10.1039/P29760001386>
- Cramer, C. J. (2004), *Essentials of computational chemistry: theories and models*, 2a edn, John Wiley & Sons Ltd, Chichester.
- Cruz-Cabeza, A. J., Reutzel-Edens, S. M. y Bernstein, J. (2015), ‘Facts and fictions about polymorphism’, *Chem. Soc. Rev.* **44**, 8619–8635.
URL: <http://dx.doi.org/10.1039/C5CS00227C>
- Dolgonos, G. A., Hoja, J. y Boese, A. D. (2019), ‘Revised values for the x23 benchmark set of molecular crystals’, *Phys. Chem. Chem. Phys.* **21**, 24333–24344.
URL: <http://dx.doi.org/10.1039/C9CP04488D>

- Ester, M., Kriegel, H.-P., Sander, J. y Xu, X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, *in* 'KDD'.
- Fleshman, W. (2019), 'Spectral clustering: foundation and application'.
URL: <https://towardsdatascience.com/spectral-clustering-aba2640c0d5b#\protect\leavevmode@ifvmode\kern+.2222em\relax~:text=Spectral%20clustering%20is%20a%20technique,non%20graph%20data%20as%20well.>
- Flower, D. (1999), 'Rotational superposition: a review of methods', *Journal of molecular graphics and modelling* **17**(3-4), 238—244.
URL: <http://europepmc.org/abstract/MED/10736782>
- Foster, M. E. y Sohlberg, K. (2010), 'Empirically corrected dft and semi-empirical methods for non-bonding interactions', *Phys. Chem. Chem. Phys.* **12**, 307–322.
URL: <http://dx.doi.org/10.1039/B912859J>
- Frey, B. J. y Dueck, D. (2007), 'Clustering by passing messages between data points', *Science* **315**(5814), 972–976.
URL: <https://www.science.org/doi/abs/10.1126/science.1136800>
- Gajda, R. y Katrusiak, A. (2011), 'Pressure-promoted ch...o hydrogen bonds in formamide aggregates', *Crystal Growth & Design* **11**(11), 4768–4774.
URL: <https://doi.org/10.1021/cg201011q>
- Geerlings, P., De Proft, F. y Langenaeker, W. (2003), 'Conceptual density functional theory', *Chem. Rev.* **103**, 1793–1873.
- Groom, C. R., Bruno, I. J., Lightfoot, M. P. y Ward, S. C. (2016), 'The Cambridge Structural Database', *Acta Crystallographica Section B* **72**(2), 171–179.
URL: <https://doi.org/10.1107/S2052520616003954>
- Grunau, C., Özüdoğru, A. A., Rozhoň, V. y Tětek, J. (2022), 'A nearly tight analysis of greedy k-means++'.
URL: <https://arxiv.org/abs/2207.07949>
- Hahn, T., ed. (2002), *International Tables for Crystallography, Volume A: Space Group Symmetry*, 5 edn, Springer Dordrecht, Dordrecht, Holland ;.
- Hanson, R. M., Kohler, D. y Braun, S. G. (2011), 'Quaternion-based definition of protein secondary structure straightness and its relationship to ramachandran angles', *Proteins: Structure, Function, and Bioinformatics* **79**(7), 2172–2180.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.23037>

- Hart, G. L. W., Jorgensen, J. J., Morgan, W. S. y Forcade, R. W. (2019), ‘A robust algorithm for k-point grid generation and symmetry reduction’, *Journal of Physics Communications* **3**(6), 065009.
URL: <https://dx.doi.org/10.1088/2399-6528/ab2937>
- Hasnip, P. J., Refson, K., Probert, M. I. J., Yates, J. R., Clark, S. J. y Pickard, C. J. (2014), ‘Density functional theory in the solid state’, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **372**(2011), 20130270.
URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2013.0270>
- Heng, T., Yang, D., Wang, R., Zhang, L., Lu, Y. y Du, G. (2021), ‘Progress in research on artificial intelligence applied to polymorphism and cocrystal prediction’, *ACS Omega* **6**(24), 15543–15550. PMID: 34179597.
URL: <https://doi.org/10.1021/acsomega.1c01330>
- Hirshfeld, F. L. (1977), ‘Bonded-atom fragments for describing molecular charge densities’, *Theoretica chimica acta* **44**, 129–138.
- Horn, B. K. P. (1987), ‘Closed-form solution of absolute orientation using unit quaternions’, *J. Opt. Soc. Am. A* **4**(4), 629–642.
URL: <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-4-4-629>
- Horn, B. K. P., Hilden, H. M. y Negahdaripour, S. (1988), ‘Closed-form solution of absolute orientation using orthonormal matrices’, *J. Opt. Soc. Am. A* **5**(7), 1127–1135.
URL: <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-5-7-1127>
- Huo, H. y Rupp, M. (2022), ‘Unified representation of molecules and crystals for machine learning’, *Machine Learning: Science and Technology* **3**(4), 045017.
URL: <https://doi.org/10.1088/2632-2153/2Faca005>
- Hönig, S. M. N., Lemmen, C. y Rarey, M. (2022), ‘Small molecule superposition: A comprehensive overview on pose scoring of the latest methods’, *WIREs Computational Molecular Science* **n/a**(n/a), e1640.
URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1640>
- Ioannidou, S. y Pantazis, G. (2020), ‘Helmert transformation problem. from euler angles method to quaternion algebra’, *ISPRS International Journal of Geo-*

Information **9**(9).

URL: <https://www.mdpi.com/2220-9964/9/9/494>

Itoh, K. y Shimanouchi, T. (1972), 'Vibrational spectra of crystalline formamide', *Journal of Molecular Spectroscopy* **42**(1), 86–99.

URL: <https://www.sciencedirect.com/science/article/pii/S0022285272901464>

Jain, A. K. (2010), 'Data clustering: 50 years beyond k-means', *Pattern Recognition Letters* **31**(8), 651–666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

URL: <https://www.sciencedirect.com/science/article/pii/S0167865509002323>

Jia, Y.-B. (s.f.), 'Dual quaternions'.

URL: <https://faculty.sites.iastate.edu/jia/files/inline-files/dual-quaternion.pdf>

Karney, C. F. (2007), 'Quaternions in molecular modeling', *Journal of Molecular Graphics and Modelling* **25**(5), 595–604.

URL: <https://www.sciencedirect.com/science/article/pii/S1093326306000829>

Katrusiak, A. (1993), 'A new polymorph of maleic hydrazide', *Acta Crystallographica Section C* **49**(1), 36–39.

URL: <https://doi.org/10.1107/S0108270192004918>

Katrusiak, A. (2001), 'Polymorphism of maleic hydrazide. I', *Acta Crystallographica Section B* **57**(5), 697–704.

URL: <https://doi.org/10.1107/S010876810101151X>

Kaufman, L. y Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction To Cluster Analysis*.

Kaxiras, E. (2003), *Atomic and Electronic Structure of Solids*, Cambridge University Press.

Kikuchi, R. (1951), 'A theory of cooperative phenomena', *Phys. Rev.* **81**, 988–1003.

URL: <https://link.aps.org/doi/10.1103/PhysRev.81.988>

Kodinariya, T. M. y Makwana, P. R. (2013), 'Review on determining number of Cluster in K-Means Clustering', *International Journal of Advance Research in Computer Science and Management Studies* **1**(6), 90–95.

- Kohn, W. y Sham, L. J. (1965), ‘Self-consistent equations including exchange and correlation effects’, *Phys. Rev.* **140**, A1133–A1138.
URL: <https://link.aps.org/doi/10.1103/PhysRev.140.A1133>
- Kratzer, P. y Neugebauer, J. (2019), ‘The basics of electronic structure theory for periodic systems’, *Frontiers in Chemistry* **7**.
URL: <https://www.frontiersin.org/articles/10.3389/fchem.2019.00106>
- Kresse, G. y Furthmüller, J. (1996), ‘Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set’, *Phys. Rev. B* **54**, 11169–11186.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.54.11169>
- Kresse, G. y Furthmüller, J. (1996), ‘Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set’, *Computational Materials Science* **6**(1), 15–50.
URL: <https://www.sciencedirect.com/science/article/pii/S0927025696000080>
- Kresse, G. y Hafner, J. (1993), ‘Ab initio molecular dynamics for liquid metals’, *Phys. Rev. B* **47**, 558–561.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.47.558>
- Kresse, G. y Joubert, D. (1999a), ‘From ultrasoft pseudopotentials to the projector augmented-wave method’, *Phys. Rev. B* **59**, 1758–1775.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.59.1758>
- Kresse, G. y Joubert, D. (1999b), ‘From ultrasoft pseudopotentials to the projector augmented-wave method’, *Phys. Rev. B* **59**, 1758–1775.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.59.1758>
- Kufareva, I. y Abagyan, R. (2012), *Methods of Protein Structure Comparison*, Humana Press, Totowa, NJ, pp. 231–257.
URL: https://doi.org/10.1007/978-1-61779-588-6_10
- Ladell, J. y Post, B. (1954a), ‘The crystal structure of formamide’, *Acta Crystallographica* **7**, 559–564.
- Ladell, J. y Post, B. (1954b), ‘The crystal structure of formamide’, *Acta Crystallographica* **7**(8-9), 559–564.
URL: <https://doi.org/10.1107/S0365110X54001855>
- Langer, M. F., Goeßmann, A. y Rupp, M. (2022), ‘Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine

- learning’, *npj Computational Materials* **8**(1).
URL: <https://doi.org/10.1038%2Fs41524-022-00721-x>
- Li, X., Curtis, F. S., Rose, T., Schober, C., Vazquez-Mayagoitia, A., Reuter, K., Oberhofer, H. y Marom, N. (2018), ‘Genarris: Random generation of molecular crystal structures and fast screening with a harris approximation’, *The Journal of Chemical Physics* **148**(24), 241701.
URL: <https://doi.org/10.1063/1.5014038>
- Lloyd, S. (1982), ‘Least squares quantization in pcm’, *IEEE Transactions on Information Theory* **28**(2), 129–137.
- Mayer, C. y Wallek, T. (2020), ‘Cluster-based thermodynamics of interacting dice in a lattice’, *Entropy* **22**(10).
URL: <https://www.mdpi.com/1099-4300/22/10/1111>
- McDonagh, D., Skylaris, C.-K. y Day, G. M. (2019), ‘Machine-learned fragment-based energies for crystal structure prediction’, *Journal of Chemical Theory and Computation* **15**(4), 2743–2758. PMID: 30817152.
URL: <https://doi.org/10.1021/acs.jctc.9b00038>
- Mechelke, M. y Habeck, M. (2010), ‘Robust probabilistic superposition and comparison of protein structures’, *BMC Bioinformatics* **11**(1), 1471–2105.
URL: <https://doi.org/10.1186/1471-2105-11-363>
- Meila, M. (2016), Spectral clustering : a tutorial for the 2010’s, *in* M. M. F. Hennig, C.; Meila y R. Rocci, eds, ‘Handbook of Cluster ANalysis’, CRC Press.
- Moreno, J. y Soler, J. M. (1992), ‘Optimal meshes for integrals in real- and reciprocal-space unit cells’, *Phys. Rev. B* **45**, 13891–13898.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.45.13891>
- Nachtergaele, Isaiah; Lankham, B. y Schilling, A. (2021), ‘The gram-schmidt orthogonalization procedure’. Online; accessed 2022-04-15.
URL: [https://math.libretexts.org/Bookshelves/Linear_Algebra/Book%3A_Linear_Algebra_\(Schilling_Nachtergaele_and_Lankham\)/09%3A_Inner_product_spaces/9.05%3A_The_Gram-Schmidt_Orthogonalization_procedure](https://math.libretexts.org/Bookshelves/Linear_Algebra/Book%3A_Linear_Algebra_(Schilling_Nachtergaele_and_Lankham)/09%3A_Inner_product_spaces/9.05%3A_The_Gram-Schmidt_Orthogonalization_procedure)
- Ng, A., Jordan, M. y Weiss, Y. (2001), On spectral clustering: Analysis and an algorithm, *in* T. Dietterich, S. Becker y Z. Ghahramani, eds, ‘Advances in Neural Information Processing Systems’, Vol. 14, MIT Press.

- URL:** <https://proceedings.neurips.cc/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf>
- Nyman, J. y Day, G. M. (2015), ‘Static and lattice vibrational energy differences between polymorphs’, *CrystEngComm* **17**, 5154–5165.
URL: <http://dx.doi.org/10.1039/C5CE00045A>
- Onat, B., Ortner, C. y Kermode, J. R. (2020), ‘Sensitivity and dimensionality of atomic environment representations used for machine learning interatomic potentials’, *The Journal of Chemical Physics* **153**(14), 144106.
URL: <https://doi.org/10.1063/5.0016005>
- Otto, S. A. (2019), ‘How to normalize the RMSE’, <https://www.marinedatascience.co/blog/2019/01/07/normalizing-the-rmse/>. Consultado: 23-12-2022.
- Parr, R. G. y Weitao, Y. (1995), *Density-Functional Theory of Atoms and Molecules*, Oxford University Press.
URL: <https://doi.org/10.1093/oso/9780195092769.001.0001>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, E. (2011), ‘Scikit-learn: Machine learning in Python’, *Journal of Machine Learning Research* **12**, 2825–2830.
- Perdew, J. P., Burke, K. y Ernzerhof, M. (1996), ‘Generalized gradient approximation made simple’, *Phys. Rev. Lett.* **77**, 3865–3868.
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.77.3865>
- Perdew, J. P. y Kurth, S. (2003), Density functionals for non-relativistic coulomb systems in the new century, in F. C., N. F. y M. M. A. L., eds, ‘A Primer in Density Functional Theory. Lecture Notes in Physics’, Vol. 620, Springer, Berlin.
- Pfattner, R., Mas-Torrent, M., Bilotti, I., Brillante, A., Milita, S., Liscio, F., Biscarini, F., Marszalek, T., Ulanski, J., Nosal, A., Gazicki-Lipman, M., Leufgen, M., Schmidt, G., Molenkamp, L. W., Laukhin, V., Veciana, J. y Rovira, C. (2010), ‘High-performance single crystal organic field-effect transistors based on two dithiophene-tetrathiafulvalene (dt-ttf) polymorphs’, *Advanced Materials* **22**(37), 4198–4203.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201001446>

- Pfleger, M., Wallek, T. y Pfennig, A. (2014), ‘Constraints of compound systems: Prerequisites for thermodynamic modeling based on shannon entropy’, *Entropy* **16**(6), 2990–3008.
URL: <https://www.mdpi.com/1099-4300/16/6/2990>
- Piaggi, P. M. y Parrinello, M. (2018), ‘Predicting polymorphism in molecular crystals using orientational entropy’, *Proceedings of the National Academy of Sciences* **115**(41), 10251–10256.
URL: <https://www.pnas.org/content/115/41/10251>
- Price, S. L. (2004), ‘The computational prediction of pharmaceutical crystal structures and polymorphism’, *Advanced Drug Delivery Reviews* **56**(3), 301–319. Pharmaceutical solid polymorphism in drug development and regulation.
URL: <https://www.sciencedirect.com/science/article/pii/S0169409X03002199>
- Price, S. L. (2008), ‘From crystal structure prediction to polymorph prediction: interpreting the crystal energy landscape’, *Phys. Chem. Chem. Phys.* **10**, 1996–2009.
URL: <http://dx.doi.org/10.1039/B719351C>
- Rappoport, D., Crawford, N. R. M., Furche, F. y Burke, K. (2009), *Approximate Density Functionals: Which Should I Choose?*, American Cancer Society.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470862106.ia615>
- Reckien, W. y Bredow, T. (2011), ‘Theoretical study of weak chemical interactions in solid formamide’, *Chemical Physics Letters* **508**(1), 54–58.
URL: <https://www.sciencedirect.com/science/article/pii/S000926141100426X>
- Reilly, A. e. a. (2016), ‘Report on the sixth blind test of organic crystal structure prediction methods’, *Acta Crystallographica Section B* **72**(4), 439–459.
URL: <https://doi.org/10.1107/S2052520616007447>
- Reilly, A. M. y Tkatchenko, A. (2013), ‘Understanding the role of vibrations, exact exchange, and many-body van der waals interactions in the cohesive properties of molecular crystals’, *The Journal of Chemical Physics* **139**(2), 024705.
URL: <https://doi.org/10.1063/1.4812819>
- Rostgaard, C. (2009), ‘The projector augmented-wave method’.
URL: <https://arxiv.org/abs/0910.1921>

Rousseeuw, P. J. (1987), ‘Silhouettes: A graphical aid to the interpretation and validation of cluster analysis’, *Journal of Computational and Applied Mathematics* **20**, 53–65.

URL: <https://www.sciencedirect.com/science/article/pii/0377042787901257>

Rupp, M., Tkatchenko, A., Müller, K.-R. y von Lilienfeld, O. A. (2012), ‘Fast and accurate modeling of molecular atomization energies with machine learning’, *Phys. Rev. Lett.* **108**, 058301.

URL: <https://link.aps.org/doi/10.1103/PhysRevLett.108.058301>

Sadeghi, A., Ghasemi, S. A., Schaefer, B., Mohr, S., Lill, M. A. y Goedecker, S. (2013), ‘Metrics for measuring distances in configuration spaces’, *The Journal of Chemical Physics* **139**(18), 184118.

URL: <https://doi.org/10.1063/1.4828704>

Sanchez, J., Ducastelle, F. y Gratias, D. (1984), ‘Generalized cluster description of multicomponent systems’, *Physica A: Statistical Mechanics and its Applications* **128**(1), 334–350.

URL: <https://www.sciencedirect.com/science/article/pii/0378437184900967>

Schubert, E., Sander, J., Ester, M., Kriegel, H. P. y Xu, X. (2017), ‘Dbscan revisited, revisited: Why and how you should (still) use dbscan’, *ACM Trans. Database Syst.* **42**(3).

URL: <https://doi.org/10.1145/3068335>

Schwartz, M. (2019), ‘Lecture 6: Entropy’.

Shannon, C. E. (1948), ‘A mathematical theory of communication’, *The Bell System Technical Journal* pp. 379–423.

Stahly, G. P. (2007), ‘Diversity in single- and multiple-component crystals. the search for and prevalence of polymorphs and cocrystals’, *Crystal Growth & Design* **7**(6), 1007–1026.

URL: <https://doi.org/10.1021/cg060838j>

Starczewski, A. y Krzyżak, A. (2015), Performance evaluation of the silhouette index, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh y J. M. Zurada, eds, ‘Artificial Intelligence and Soft Computing’, Springer International Publishing, Cham, pp. 49–58.

- Szabo, A. y Ostlund, N. S. (1989), *Modern quantum chemistry: introduction to advanced electronic structure theory*, McGraw-Hill, Inc., New York.
- Thalladi, V. R., Nüsse, M. y Boese, R. (2000), ‘The melting point alternation in α,ω -alkanedicarboxylic acids’, *Journal of the American Chemical Society* **122**(38), 9227–9236.
URL: <https://doi.org/10.1021/ja0011459>
- Thavikulwat, P. (2008), ‘Affinity propagation: a clustering algorithm for computer-assisted business simulations and experiential exercises’, *Developments in Business Simulation and Experiential Learning* **35**.
- Tom, R., Rose, T., Bier, I., O’Brien, H., Álvaro Vázquez-Mayagoitia y Marom, N. (2020), ‘Genarris 2.0: A random structure generator for molecular crystals’, *Computer Physics Communications* **250**, 107170.
URL: <https://www.sciencedirect.com/science/article/pii/S0010465520300242>
- van de Streek, J. y Neumann, M. A. (2010), ‘Validation of experimental molecular crystal structures with dispersion-corrected density functional theory calculations’, *Acta Crystallographica Section B* **66**(5), 544–558.
URL: <https://doi.org/10.1107/S01087681100031873>
- Vinograd, V. L. y Perchuk, L. L. (1996), ‘Informational models for the configurational entropy of regular solid solutions: flat lattices’, *The Journal of Physical Chemistry* **100**(39), 15972–15985.
URL: <https://doi.org/10.1021/jp960416s>
- von Luxburg, U. (2007), ‘A tutorial on spectral clustering’.
URL: <https://arxiv.org/abs/0711.0189>
- Wallek, T., Pfleger, M. y Pfennig, A. (2016), ‘Discrete modeling of lattice systems: The concept of shannon entropy applied to strongly interacting systems’, *Industrial & Engineering Chemistry Research* **55**(8), 2483–2492.
URL: <https://doi.org/10.1021/acs.iecr.5b04430>
- Wisesa, P., McGill, K. A. y Mueller, T. (2016), ‘Efficient generation of generalized monkhorst-pack grids through the use of informatics’, *Phys. Rev. B* **93**, 155109.
URL: <https://link.aps.org/doi/10.1103/PhysRevB.93.155109>
- Wöhler y Liebig (1832), ‘Untersuchungen über das radikal der benzoessäure’, *Annalen der Pharmacie* **3**(3), 249–282.

URL: <https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/jlac.18320030302>

Xu, D. y Tian, Y. (2015), ‘A Comprehensive Survey of Clustering Algorithms’, *Annals of Data Science* **2**(2), 165–193.

URL: <https://doi.org/10.1007/s40745-015-0040-1>

Yan, D. y Evans, D. G. (2014), ‘Molecular crystalline materials with tunable luminescent properties: from polymorphs to multi-component solids’, *Mater. Horiz.* **1**, 46–57.

URL: <http://dx.doi.org/10.1039/C3MH00023K>

Zeng, H., Fang, X., Chang, G. y Yang, R. (2018), ‘A dual quaternion algorithm of the helmert transformation problem’, *Earth, Planets and Space* **70**(26), 1880.

URL: <https://doi.org/10.1186/s40623-018-0792-x>

9. Anexo A

9.1. Análisis adicionales

Análisis de las métricas

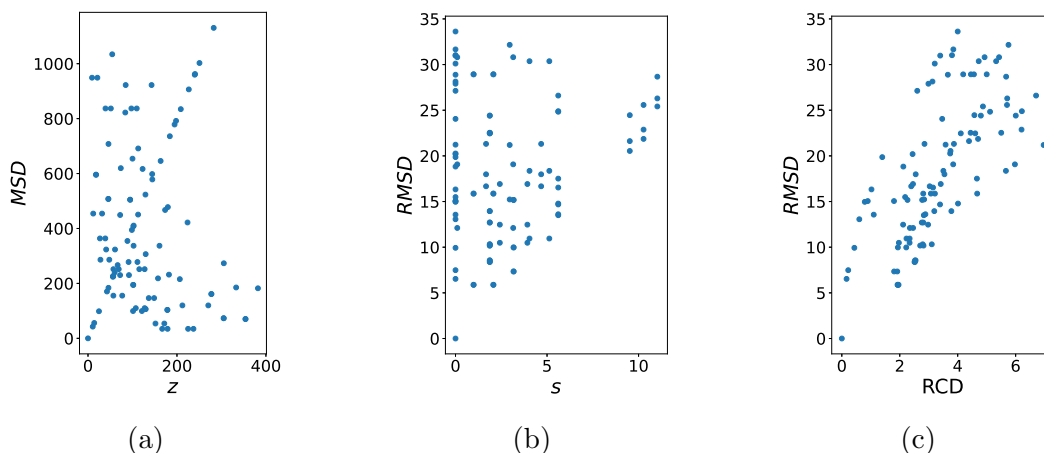


Figura 30: $RMSD$ de 106 moléculas de una de las estructuras de la **hidrazida maleica** respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en relación con las métricas (a) z , (b) s y (c) RCD de las mismas moléculas. En el caso de la gráfica (a) se muestra el valor cuadrado del $RMSD$, sea el MSD , con el fin de mostrar de manera más clara la relación entre esta medida y z .

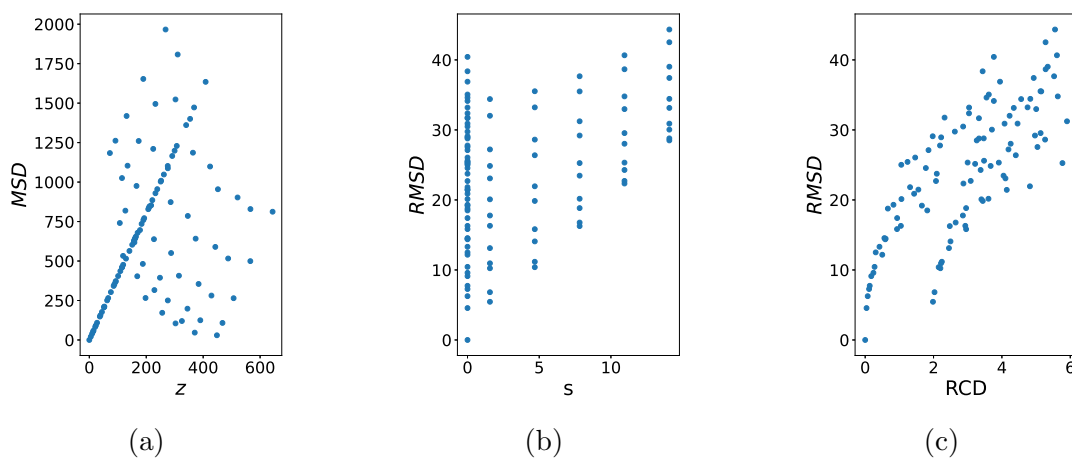


Figura 31: $RMSD$ de 106 moléculas de una de las estructuras del **ácido oxálico** respecto a la primera molécula en la misma estructura (sin considerar la molécula de referencia) en el mismo sistema en relación con las métricas (a) z , (b) s y (c) RCD de las mismas moléculas. En el caso de la gráfica (a) se muestra el valor cuadrado del $RMSD$, sea el MSD , con el fin de mostrar de manera más clara la relación entre esta medida y z .

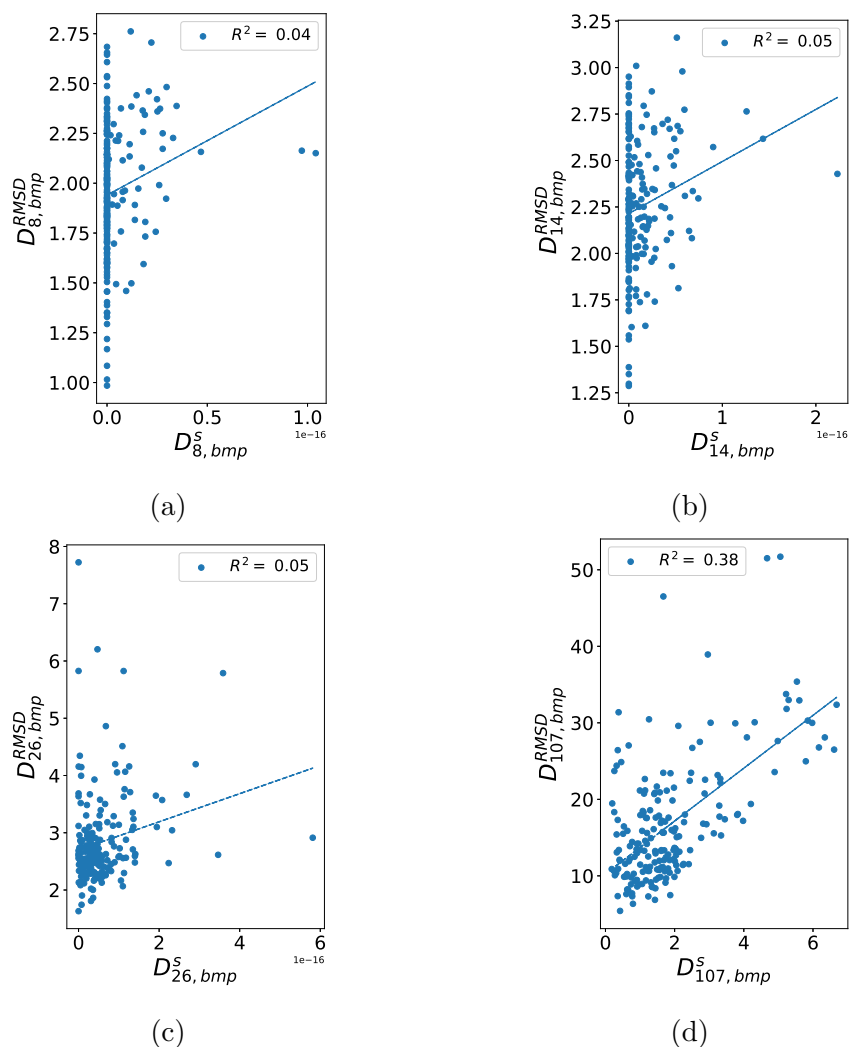


Figura 32: Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^s$ obtenidos con el algoritmo *bmp* para (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia (l). Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos.

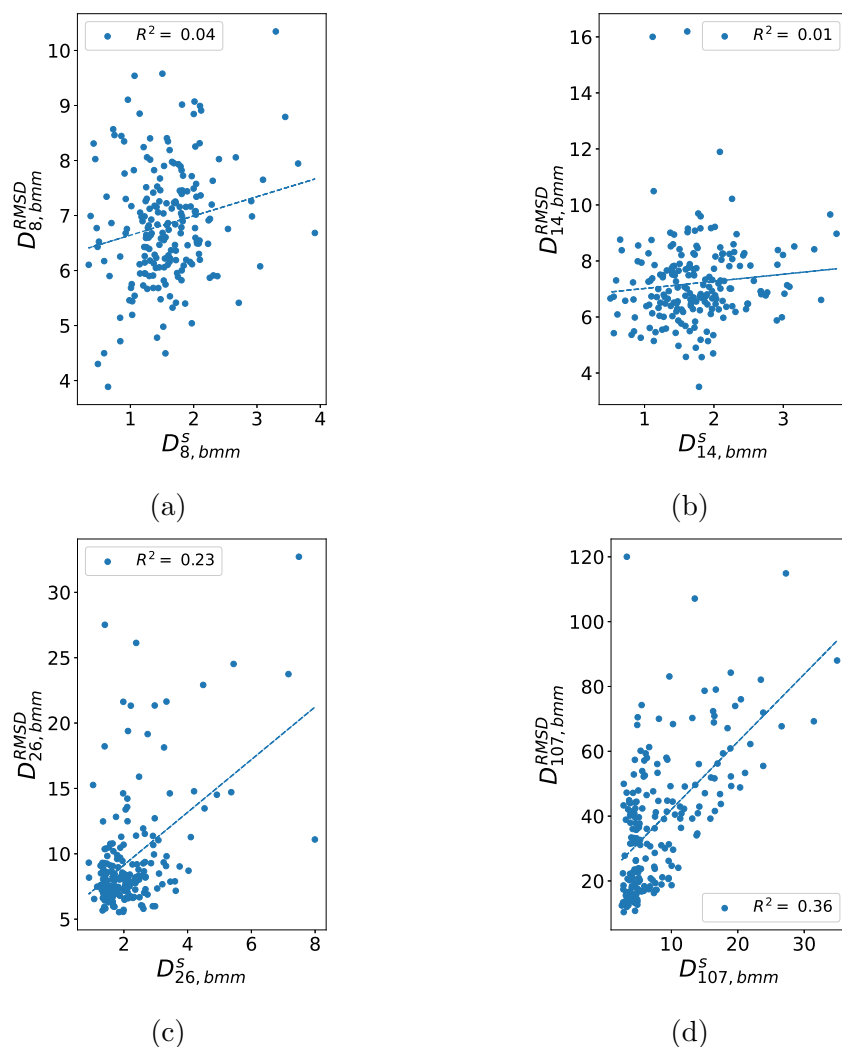


Figura 33: Gráficas de los valores en el primer renglón de la matriz de distancia $\mathbf{D}^{MSD}$ contra los valores correspondientes de la matriz $\mathbf{D}^s$ obtenidos con el algoritmo *bmm* para (a) 8, (b) 14, (c) 26 y (d) 107 moléculas vecinas (l) en relación a la molécula de referencia (l). Para todas las matrices se consideró un mismo subconjunto de 200 estructuras de la formamida. En cada gráfica se muestra el valor de R^2 de la regresión lineal de los datos.

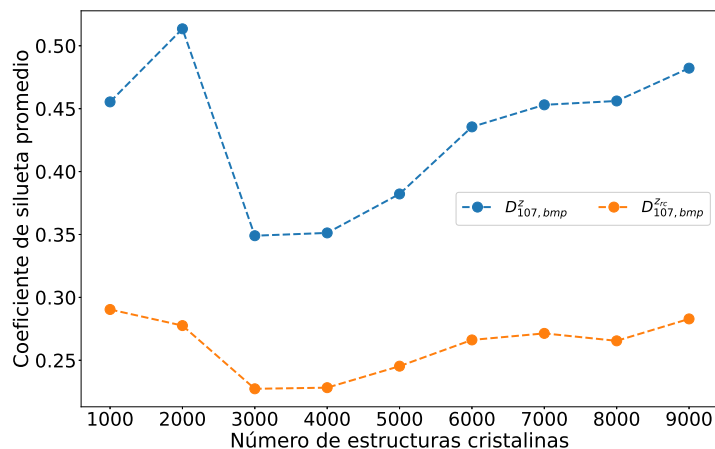


Figura 34: Coeficiente de silueta promedio de los clústeres obtenidos con la medida z y su raíz cuadrada (z_{rc}) considerando 107 moléculas vecinas a la molécula de referencia (l) al calcular las matrices de distancia con el algoritmo *bmp* para subconjuntos de diferentes tamaños de estructuras cristalinas de la formamida; en todos los casos se usó el método de agrupamiento espectral.

De acuerdo con la figura 34, se decidió descartar la medida z_{rc} , ya que esta genera para todos los subconjuntos de estructuras cristalinas los valores más bajos del coeficiente de silueta promedio. Este análisis se realizó en una etapa relativamente temprana del proyecto y, por lo tanto, no se extendió el análisis de la medida z_{rc} . Aun así, tanto z como su raíz cuadrada se establecieron bajo la misma metodología y, por lo tanto, representan conceptualmente la misma métrica entre las orientaciones relativas de pares de moléculas. Así, aunque ambas métricas llevan a diferentes agrupaciones, dado que no están relacionadas de manera lineal, se decidió enfocar el análisis únicamente en z considerando que esta lleva a los mejores clústeres.

Resultados para las estructuras cristalinas de *MH* y *OA*

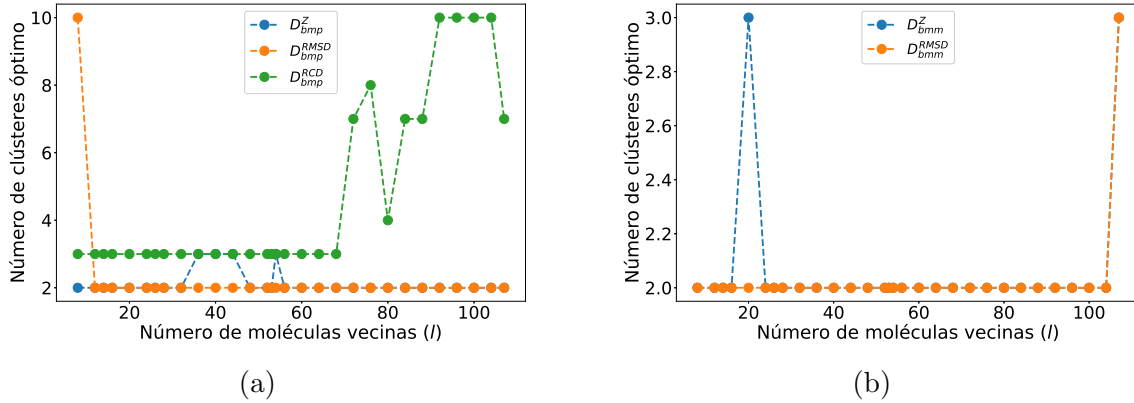


Figura 35: Número de clústeres óptimos obtenidos con las medidas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 1092 estructuras cristalinas de la hidrazida maleica (*MH*); en todos los casos se usó el método de agrupamiento espectral.

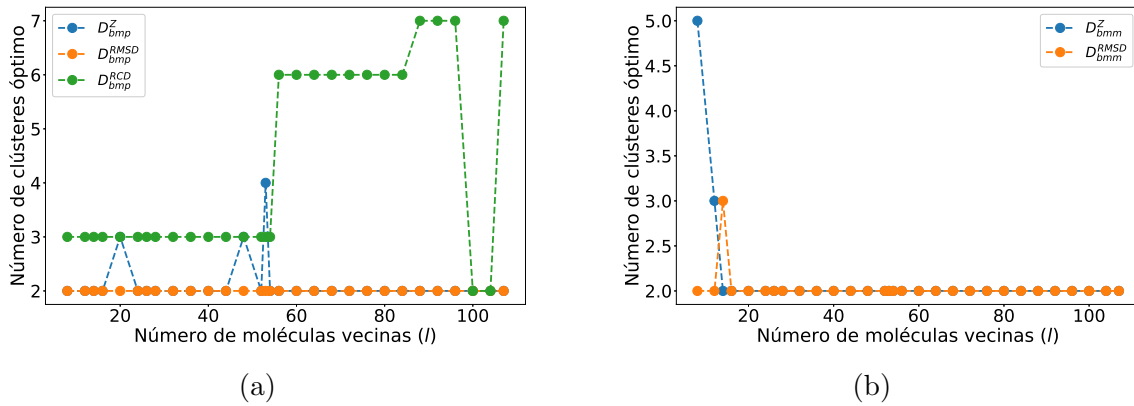


Figura 36: Número de clústeres óptimos obtenidos con las medidas z , $RMSD$ y RCD para diferentes valores del parámetro l al calcular las matrices de distancia con los algoritmos (a) *bmp* y (b) *bmm* para un conjunto de 970 estructuras cristalinas del ácido oxálico (*OA*); en todos los casos se usó el método de agrupamiento espectral.

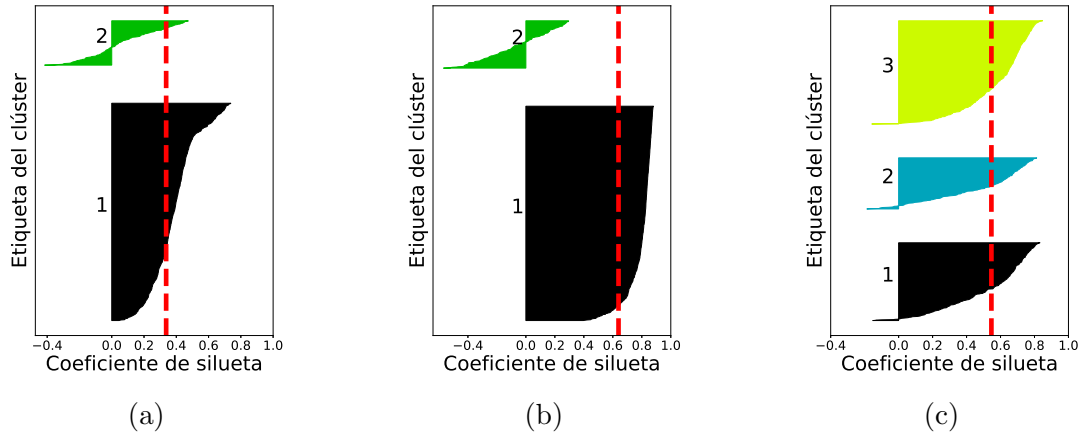


Figura 37: Coeficientes de silueta para cada estructura en los clústeres obtenidos con las matrices de distancia (a) $D_{107,bmp}^z$, (b) $D_{107,bmm}^z$ y (c) D_8^{RCD} , de un conjunto de 970 estructuras cristalinas del ácido oxálico (OA). El coeficiente de silueta promedio se muestra con la línea roja discontinua.

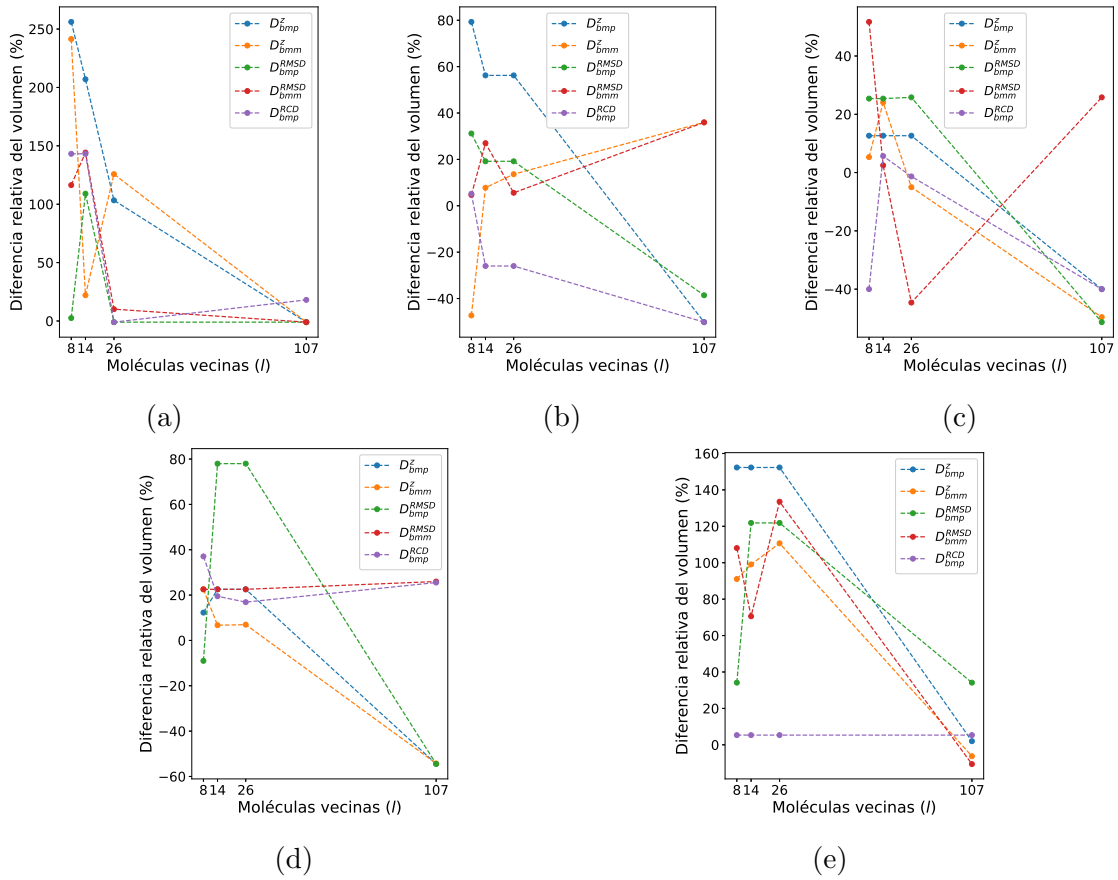


Figura 38: Diferencia relativa en porcentaje entre el volumen de la celda unitaria de las estructuras determinadas experimentalmente (a) $MH1$, (b) $MH2$, (c) $MH3$, (d) $OA - \alpha$ y (e) $OA - \beta$ y las estructuras cristalinas identificadas como similares con las métricas z , $RMSD$ y RCD . En el caso de las medidas z y $RMSD$ se emplearon los algoritmos bmp y bmm . Todas las matrices de distancia se calcularon considerando $l = 8, 14, 26$ y 107 .

10. Anexo B

10.1. Demostración del cambio de referencia

En esta demostración se usará la letra v para los vectores en una estructura cristalina α y la letra u para aquellos en una estructura cristalina β . Se usarán las letras primadas en caso de que ocurra una transformación de los vectores sin un cambio del marco de referencia. Asimismo, los subíndices i denotan cuaterniones duales con molécula de referencia en la estructura α , mientras que aquellos con subíndices j se definen respecto a la referencia β . Así, se propone el parámetro de comparación

$$\sigma_d = \sigma_i \otimes \sigma_j'^*, \quad (115)$$

tal que

$$\sigma_j' = \sigma_{\alpha\beta}^{R*} \sigma_j^R \sigma_{\alpha\beta}^R \quad (116)$$

y

$$\sigma_{\alpha\beta}^R = q_{\alpha\beta}^R + \epsilon(0, 0, 0, 0), \quad (117)$$

sea $q_{\alpha\beta}^R$ el cuaternión que rota la molécula de referencia α a la orientación de la molécula de referencia β .

Así, sea un vector $\vec{v}$ bajo la base α , tal que:

$$\sigma_d(1 + \epsilon v)\sigma_d^\diamond \quad t.q. \quad v = (0, \vec{v}) \quad (118)$$

$$= \sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^* \otimes \sigma_{\alpha\beta}^R (1 + \epsilon v) (\sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^* \otimes \sigma_{\alpha\beta}^R)^\diamond \quad (119)$$

$$= \sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^* \otimes \sigma_{\alpha\beta}^R (1 + \epsilon v) \sigma_{\alpha\beta}^{R\diamond} (\sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^*)^\diamond \quad (120)$$

$$= \sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^* \otimes q_{\alpha\beta}^R (1 + \epsilon v) q_{\alpha\beta}^{R*} (\sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^*)^\diamond \quad (121)$$

$$= \sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^* \otimes (1 + \epsilon q_{\alpha\beta}^R v q_{\alpha\beta}^{R*}) (\sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^*)^\diamond \quad (122)$$

$$sea \quad q_{\alpha\beta}^R v q_{\alpha\beta}^{R*} = (1 + \epsilon u) \quad con \quad u = (0, \vec{u}) \quad (123)$$

donde $\vec{u}$ es el vector $\vec{v}$ con respecto a la molécula de referencia en β

$$\sigma_d(1 + \epsilon v)\sigma_d^\diamond = \sigma_i \otimes \sigma_{\alpha\beta}^R \otimes \sigma_j^*(1 + \epsilon u)\sigma_j^{*\diamond}(\sigma_i \otimes \sigma_{\alpha\beta}^R)^\diamond \quad (124)$$

$$= \sigma_i \otimes q_{\alpha\beta}^R(1 + \epsilon u')q_{\alpha\beta}^{R*} \otimes \sigma_i^\diamond \quad (125)$$

$$= \sigma_i \otimes (1 + \epsilon v') \otimes \sigma_i^\diamond \quad (126)$$

$$= 1 + \epsilon v'' \quad (127)$$

tal que v'' se encuentra en el espacio de la estructura α .

Una demostración alternativa consiste en considerar que al cambiar de referencia un cuaternión dual solo deben modificarse las orientaciones de su vector de rotación $\hat{t}$ y de su vector de traslación $\vec{T}$, ya que las moléculas de referencia difieren únicamente en sus orientaciones. Así, el ángulo de rotación y la magnitud de $\vec{T}$ deben permanecer inalterados con el fin de representar la misma transformación. Con este fin, se analiza primero el efecto de los cuaterniones duales de cambio de referencia, $\sigma_{\alpha\beta}^R$, sobre un cuaternión (cuaternión dual cuya parte dual es nula), p:

$$\begin{aligned} \sigma_{\alpha\beta}^{R*} \otimes (p + \epsilon(0, 0, 0, 0)) \otimes \sigma_{\alpha\beta}^R &= q_{\alpha\beta}^{R*} p q_{\alpha\beta}^R \\ &= q_{\alpha\beta}^{R*}(p_0, \vec{p}) q_{\alpha\beta}^R \\ &= q_{\alpha\beta}^{R*}(p_0, \vec{p})(c_{\alpha\beta}, \hat{t}_{\alpha\beta} s_{\alpha\beta}), \end{aligned}$$

donde $t_{\alpha\beta}$ es el vector de rotación entre los marcos de referencia, $c_{\alpha\beta} = \cos(\theta_{\alpha\beta}/2)$, $s_{\alpha\beta} = \sin(\theta_{\alpha\beta}/2)$, sea $\theta_{\alpha\beta}$ el ángulo de rotación.

$$\begin{aligned} q_{\alpha\beta}^{R*} p q_{\alpha\beta}^R &= q_{\alpha\beta}^{R*}(p_0, \vec{p})(c_{\alpha\beta}, \hat{t}_{\alpha\beta} s_{\alpha\beta}) \\ &= q_{\alpha\beta}^{R*}(p_0 c_{\alpha\beta} - s_{\alpha\beta} \vec{p} \cdot \hat{t}_{\alpha\beta} + p_0 s_{\alpha\beta} \hat{t}_{\alpha\beta} + c_{\alpha\beta} \vec{p} + s_{\alpha\beta} \vec{p} \times \hat{t}_{\alpha\beta}) \\ &= (c_{\alpha\beta}, -\hat{t}_{\alpha\beta} s_{\alpha\beta}) (p_0 c_{\alpha\beta} - s_{\alpha\beta} \vec{p} \cdot \hat{t}_{\alpha\beta}, p_0 s_{\alpha\beta} \hat{t}_{\alpha\beta} + c_{\alpha\beta} \vec{p} + s_{\alpha\beta} \vec{p} \times \hat{t}_{\alpha\beta}) \\ &= p_0 c_{\alpha\beta}^2 - c_{\alpha\beta} s_{\alpha\beta} \vec{p} \cdot \hat{t}_{\alpha\beta} + p_0 s_{\alpha\beta}^2 \hat{t}_{\alpha\beta} \cdot \hat{t}_{\alpha\beta} + c_{\alpha\beta} s_{\alpha\beta} \hat{t}_{\alpha\beta} \cdot \vec{p} + s_{\alpha\beta}^2 \hat{t}_{\alpha\beta} \cdot (\vec{p} \times \hat{t}_{\alpha\beta}) \\ &\quad + p_0 c_{\alpha\beta} s_{\alpha\beta} \hat{t}_{\alpha\beta} + c_{\alpha\beta}^2 \vec{p} + c_{\alpha\beta} s_{\alpha\beta} \vec{p} \times \hat{t}_{\alpha\beta} \\ &\quad - p_0 c_{\alpha\beta} s_{\alpha\beta} \hat{t}_{\alpha\beta} + s_{\alpha\beta}^2 (\vec{p} \cdot \hat{t}_{\alpha\beta}) \hat{t}_{\alpha\beta} \\ &\quad - p_0 s_{\alpha\beta}^2 \hat{t}_{\alpha\beta} \times \hat{t}_{\alpha\beta} - c_{\alpha\beta} s_{\alpha\beta} \hat{t}_{\alpha\beta} \times \vec{p} - s_{\alpha\beta}^2 \hat{t}_{\alpha\beta} \times (\vec{p} \times \hat{t}_{\alpha\beta}) \\ &= p_0 (c_{\alpha\beta}^2 + s_{\alpha\beta}^2) + c_{\alpha\beta}^2 \vec{p} + s_{\alpha\beta}^2 (\vec{p} \cdot \hat{t}_{\alpha\beta}) \hat{t}_{\alpha\beta} + 2c_{\alpha\beta} s_{\alpha\beta} \vec{p} \times \hat{t}_{\alpha\beta} \\ &= (p_0, c_{\alpha\beta}^2 \vec{p} + s_{\alpha\beta}^2 (\vec{p} \cdot \hat{t}_{\alpha\beta}) \hat{t}_{\alpha\beta} + 2c_{\alpha\beta} s_{\alpha\beta} \vec{p} \times \hat{t}_{\alpha\beta}) \end{aligned}$$

Entonces, el cuaternión dual de cambio de base no tiene ningún efecto sobre la parte escalar del cuaternión sobre el que actúa. Por lo tanto,

$$q_{\alpha\beta}^{R*} p q_{\alpha\beta}^R = (p_0, \vec{0}) + q_{\alpha\beta}^{R*}(0, \vec{p}) q_{\alpha\beta}^R = (p_0, \vec{0}) + q_{\beta\alpha}^R(0, \vec{p}) q_{\beta\alpha}^{R*}$$

Es decir, el cuaternión dual $\sigma_{\alpha\beta}$ rota la componente vectorial del cuaternión sobre el que actúa. En consecuencia, se tiene lo siguiente al aplicar el cuaternión dual de cambio de base sobre un cuaternión dual que representa a su vez una transformación de cuerpo rígido:

$$\begin{aligned} \sigma_{\alpha\beta}^{R*} \sigma_j^R \sigma_{\alpha\beta}^R &= \sigma_{\alpha\beta}^{R*} \otimes (q_j^R + \frac{1}{2} \epsilon q_j^T q_j^R) \otimes \sigma_{\alpha\beta}^R \\ &= q_{\alpha\beta}^{R*} \otimes (q_j^R + \frac{1}{2} \epsilon q_j^T q_j^R) \otimes q_{\alpha\beta}^R \\ &= q_{\alpha\beta}^{R*} q_j^R q_{\alpha\beta}^R + \frac{1}{2} \epsilon q_{\alpha\beta}^{R*} q_j^T q_j^R q_{\alpha\beta}^R \\ &= q_{\alpha\beta}^{R*} q_j^R q_{\alpha\beta}^R + \frac{1}{2} \epsilon q_{\alpha\beta}^{R*} q_j^T q_{\alpha\beta}^R q_{\alpha\beta}^{R*} q_j^R q_{\alpha\beta}^R \\ &= (c_j, \vec{0}) + q_{\alpha\beta}^{R*}(0, \hat{t}_j s_j) q_{\alpha\beta}^R + \frac{1}{2} \epsilon [q_{\alpha\beta}^{R*}(0, \vec{T}) q_{\alpha\beta}^R] [(c_j, 0) + q_{\alpha\beta}^{R*}(0, \hat{t}_j s_j) q_{\alpha\beta}^R] \\ &= (c_j, \vec{0}) + s_j q_{\alpha\beta}^{R*}(0, \hat{t}_j) q_{\alpha\beta}^R + \frac{1}{2} \epsilon [q_{\alpha\beta}^{R*}(0, \vec{T}) q_{\alpha\beta}^R] [(c_j, 0) + s_j q_{\alpha\beta}^{R*}(0, \hat{t}_j) q_{\alpha\beta}^R] \end{aligned}$$

De esta manera, el cuaternión dual de cambio de referencia, $\sigma_{\alpha\beta}^R$, actúa únicamente sobre los vectores de rotación, $\hat{t}$, y de posición, $\vec{T}$. Por lo tanto, el cambio de base propuesto es adecuado. Esto es, este no modifica la transformación representada por los cuaterniones duales, solo modifica la orientación de la misma con el fin de que esta quede referida en el nuevo marco de referencia.

10.2. Demostración de la simetría de las medidas s y z

De acuerdo con las propiedades de las métricas, estas deben ser simétricas; en otras palabras, deben ser invariantes ante al orden de comparación de las moléculas. En el caso de la medidas s y z , la simetría se cumple de acuerdo con las siguientes propiedades de los cuaterniones duales:

$$\sigma_i \otimes \sigma_j^* = (\sigma_j \otimes \sigma_i^*)^* \quad (128)$$

Por lo tanto, considerando solo la componente dual:

$$q_{ij}^{T,r} = q_{ji}^{T,r*} \quad (129)$$

Así, dado que el conjugado de un cuaternión solo afecta a la parte vectorial:

$$s_{ij} = \text{escalar}(q_{ij}^{T,r}) \quad (130)$$

$$= \text{escalar}(q_{ji}^{T,r*}) \quad (131)$$

$$= \text{escalar}(q_{ji}^{T,r}) \quad (132)$$

$$= s_{ji} \quad (133)$$

De manera similar se tiene para la medida z :

$$z_{ij} = |q_{ij}^{T,r}| \quad (134)$$

$$= |q_{ji}^{T,r*}| \quad (135)$$

$$= |q_{ji}^{T,r}| \quad (136)$$

$$= z_{ji} \quad (137)$$

Así, aunque el producto de cuaterniones duales no es conmutativo, las métricas propuestas (s y z) sí son simétricas. Sin embargo, las matrices resultantes (Z) no son simétricas, ya que por contrucción los cuaterniones comparados pertenecen a estructuras diferentes. Así que lo anterior se cumple solo en el caso de que se compare una estructura consigo misma. En el caso general se tiene lo siguiente:

$$z_{ij} = z_{ij}^t \quad (138)$$

donde t indica la matriz transpuesta. Esto implica que:

$$Z^{\alpha,\beta} = (Z^{\beta,\alpha})^t \quad (139)$$

Así, las matrices no son simétricas; sin embargo, al implementar los algoritmos de búsqueda que condensan las matrices de distancia en un único valor se llega a valores simétricos, ya que las búsquedas son globales y, por lo tanto, son indiferentes ante el orden de los renglones y columnas:

$$D_{\alpha\beta,bmp} = \frac{1}{l} \sum_{k=1}^l bm(Z^{\alpha,\beta}) = \frac{1}{l} \sum_{k=1}^l bm(Z^{\beta,\alpha t}) = \frac{1}{l} \sum_{k=1}^l bm(Z^{\beta,\alpha}) = D_{\beta\alpha,bmp}, \quad (140)$$

donde bm es cualquiera de los algoritmos de búsqueda de mínimos (bmp o bmm).

Caso particular de la medida s

En la sección de resultados de este proyecto se mostró que la medida s tiende a valer cero en muchas de las comparaciones de pares de moléculas, lo cual se debe a que no existe un cambio de orientación entre estas. De acuerdo con lo presentado en la metodología, para un par de moléculas se tiene lo siguiente:

$$\sigma_i^\alpha \otimes \sigma_j^{\alpha*} = (q_i^R + \epsilon q_i^{T,R}) \otimes (q_j^R + \epsilon q_j^{T,R})^* \quad (141)$$

$$= q_i^R \otimes q_j^{R*} + \epsilon(q_i^R \otimes q_j^{T,R*} + q_i^{T,R} \otimes q_j^{R*}) \quad (142)$$

Si las moléculas tienen la misma orientación, $q_i^R = q_j^R$, tal que $q_i^R \otimes q_j^{R*} = (1, 0, 0, 0)$. Así, desarrollando $q^{T,R}$:

$$\begin{aligned} \sigma_i^\alpha \otimes \sigma_j^{\alpha*} &= \mathbf{1} + \epsilon \left[q_i^R \otimes \left(\frac{1}{2} q_j^T q_j^R \right)^* + \left(\frac{1}{2} q_i^T q_i^R \right) \otimes q_j^{R*} \right] \\ &= \mathbf{1} + \frac{1}{2} \epsilon (q_i^R \otimes q_j^{R*} \otimes q_j^{T*} + q_i^T \otimes q_i^R \otimes q_j^{R*}) \\ &= \mathbf{1} + \frac{1}{2} \epsilon (q_j^{T*} + q_i^T) \\ &= \mathbf{1} + \epsilon \left[0, \frac{1}{2} (\vec{T}_i - \vec{T}_j) \right]. \end{aligned}$$

Tal que que la diferencia entre las moléculas depende únicamente de la diferencia entre los vectores de posición. Por lo tanto, como s es el valor absoluto de la parte escalar del término dual:

$$s_{ij} = 0 \quad (143)$$

11. Anexo C

11.1. Código para la generación de matrices de distancia

11.1.1. Cálculo de las medidas z y s

```
1      PROGRAM COMPARE STRUCTURES Z
2
3      IMPLICIT NONE
4
5      REAL, ALLOCATABLE :: dquaternions(:,:,:), comparison_z(:,:),
6      $                    minimums_test(:), dm_z(:,:), minimums(:),
7      $                    minimums_col(:), dq2(:,:), pq(:,:), qp(:,:),
8      $                    minimums_row(:), qresult(:,:), qtmp(:,:),
9      $                    dqref(:,:,:), ref_molecules(:,:,:), T_ref
10     (:,:)
11     INTEGER, ALLOCATABLE :: min_loc(:), structure_size(:)
12     LOGICAL, ALLOCATABLE :: mask_search(:,:), mask(:)
13     REAL dq_tmp(8), dq1(8), maxval1, maxval2, start, finish,
14     $     ref_molecule(3,4), dqrefcj(8), t1
15     CHARACTER comment, name_z*50, name_s*50, auxc*30, name_t*100
16     INTEGER initial_struct, final_struct, combinations, count_,
17     $     n_structures, truncate, n_tmp, n_molecules_l(30),
18     $     n_molecules_s, max_size_l(30), n, m, dq_limit
19     INCLUDE 'common.h'
20
21     CALL CPU_TIME(start)
22
23     OPEN(101, FILE='input')
24
25     !LECTURA DEL ARCHIVO DE ENTRADA
26     READ(101,*) comment, initial_struct, final_struct
27     READ(101,*) comment, n_molecules_s
28     DO i=1,n_molecules_s
29         READ(101,*) n_molecules_l(i)
30     ENDDO
31     READ(101,*) comment, n_atoms
32     READ(101,*) comment, start_zone
33     READ(101,*) comment, truncate
34     READ(101,*) comment, min_limit
35     READ(101,*) comment, algorithm
36
37     CLOSE(101)
38     !TERMINA LECTURA DEL ARCHIVO DE ENTRADA
```

```

39 !SE DETERMINA EL NUMERO DE ESTRUCTURAS A ANALIZAR Y EL NUMERO MAXIMO DE
40 !CUATERNIONES DUALES A GENERAR PARA CADA ESTRUCTURA
41     n_structures = final_struct-initial_struct+1
42
43     IF (min_limit == 0) THEN
44         zone = start_zone
45         max_size = MAXVAL(n_molecules_l(1:n_molecules_s))
46         $             *(zone*2+1)**3-1
47         min_limit = max_size
48     ELSE
49         DO i=1,n_molecules_s
50             zone = start_zone
51             max_size = n_molecules_l(i)*(zone*2+1)**3-1!No considerar la
referencia consigo misma
52             DO WHILE (max_size<min_limit) !Se busca que el numero de
53 !cuaterniones duales para cada estructura sea de al menos min_limit
54                 zone = zone+1
55                 max_size = n_molecules_l(i)*(zone*2+1)**3-1
56             ENDDO
57             max_size_l(i) = max_size
58         ENDDO
59         max_size = MAXVAL(max_size_l(:n_molecules_s))
60     ENDIF
61
62     ALLOCATE(dquaternions(n_structures,8,max_size),
63     $         comparison_z(max_size,max_size), minimums_col(max_size),
64     $         dm_z(n_structures,n_structures), minimums_row(max_size),
65     $         dq2(8,max_size),qresult(4,max_size), qtmp(4,max_size),
66     $         pq(4,max_size), qp(4,max_size), minimums(max_size),
67     $         minimums_test(max_size), min_loc(max_size),
68     $         mask_search(max_size,max_size),
69     $         ref_molecules(n_structures,3,4), T_ref(3,n_structures),
70     $         dqref(n_structures,8,n_structures), mask(max_size),
71     $         structure_size(n_structures),
72     $         STAT=estat)
73
74     IF (estat/=0) PRINT*, 'ERROR DURING ALLOCATION IN MAIN'
75
76 !CALCULO DE CUATERNIONES DUALES DE LAS MOLECULAS EN CADA ESTRUCTURA
77     count_ = 1
78     IF (algorithm == 2) THEN !Se asigna el mismo numero de
79 ! cuaterniones duales para todas las estructuras en el caso del segundo
80 ! algoritmo.
81         dq_limit = min_limit
82         structure_size = min_limit

```

```

83      ELSE
84          dq_limit = max_size
85      ENDIF
86      DO i=initial_struct , final_struct
87          CALL GENERATEMAIN(i,1,dquaternions(count_,1:8,1:dq_limit) ,
88              $                      ref_molecules(count_,1:3,1:4))
89      !EN ref_molecules SE ALMACENA EL CENTRO GEOMETRICO Y LOS TRES VECTORES
90      !ORTONORMALES, EN ESE ORDEN, DE LA MOLECULA DE REFERENCIA EN
91      !CADA ESTRUCTURA.
92          IF (algorithm /= 2) structure_size(count_) = size_ !Se guarda
          el
93      ! numero de cuaterniones duales de cada estructura
94          count_ = count_+1
95      ENDDO
96      !-----
97
98      !SE CALCULAN LOS CUATERNIONES DUALES ENTRE MOLECULAS DE REFERENCIA PARA
99      !CADA PAR DE ESTRUCTURAS.
100      DO i=1,n_structures
101          ref_molecule(1:3,1:4) = ref_molecules(i,1:3,1:4)
102      DO k=i , n_structures
103          T_ref(1:3,k) =
104      $          ref_molecules(k,1:3,1)-ref_molecule(1:3,1)
105      ENDDO
106      n_tmp = n_structures-i+1
107      !!SE GENERA LA PARTE ROTACIONAL DE LOS CUATERNIONES DUALES
108      CALL CALC_QUATERNIONS(
109      $          ref_molecules(i:n_structures,1:3,2:4) ,
110      $          ref_molecule(1:3,2:4) ,dqref(i,1:4,i:n_structures) ,n_tmp ,
111      $          T_ref(1:3,i:n_structures))
112      !
113      dqref(i,5:8,i:n_structures) = 0.
114      !
115      !! VERIFICACION DE LOS CUATERNIONES GENERADOS (OPCIONAL)
116      IF (i==1) THEN
117          CALL VERIFY_QUAT2(ref_molecules(i:n_structures,1:3,2:4) ,
118      $          ref_molecule(1:3,2:4) ,dqref(i,1:4,i:n_structures) ,
119      $          T_ref(1:3,i:n_structures) ,n_tmp,3)
120      !! SUBROUTINE VERIFY_QUAT(coor , coor_ref , quaternions , gcenters , n,m)
121      ENDIF
122      ENDDO
123      !!TERMINA CALCULO DE CUATERNIONES DUALES ENTRE REFERENCIAS
124      !
125      CALL CPU_TIME(finish)
126      !

```

```

126      t1=finish-start !Tiempo en generar cuaterniones duales
127
128      CALL CPU_TIME(start)
129      IF (n_structures > 1) THEN
130 !FORMATO DEL NOMBRE DEL ARCHIVO DE SALIDA
131      WRITE(auxc,'(I8)') initial_struct
132      name_z = 'dm_z_'//TRIM(ADJUSTL(auxc))//'_-'
133      name_s = 'dm_z_'//TRIM(ADJUSTL(auxc))//'_-'
134      IF (truncate /= 0) THEN
135          WRITE(auxc,'(I8)') truncate
136      ELSE
137          WRITE(auxc,'(I8)') final_struct
138      ENDIF
139      name_z = TRIM(name_z)//TRIM(ADJUSTL(auxc))//'.out'
140      name_t = TRIM(name_s)//TRIM(ADJUSTL(auxc))//'time.out'
141      OPEN(101,FILE=name_z)
142      OPEN(106,FILE=name_t)
143      WRITE(106,*) 'DUAL QUATERNIONS TIME:', t1
144      count_ = 1
145 !COMIENZA COMPARACION DE ESTRUCTURAS
146      DO l=1,n_structures
147          m = structure_size(l)
148          DO k=1,n_structures
149              n = structure_size(k)
150 !Solo en el caso del algoritmo bmm (2) m=n
151 !TRANSFORMACION DE LOS CUATERNIONES DE LA ESTRUCTURA k A LA REFERENCIA
    DE LA ESTRUCTURA l
152          dq2(1:8,1:n) = dquaternions(k,1:8,1:n)
153          dqrefcj(1:8) = dqref(l,1:8,k)
154          dqrefcj(2:4) = -1.*dqrefcj(2:4)
155          dqrefcj(6:8) = -1.*dqrefcj(6:8)
156          DO i=1,n
157              CALL DQPROD(dq2(1:8,i),dqref(l,1:8,k),dq_tmp(1:8))
158              CALL DQPROD(dqrefcj(1:8),dq_tmp(1:8),dq2(1:8,i))
159          ENDDO
160 !TERMINA TRANSFORMACION
161 !!Aqui se utiliza el segundo conjugado del segundo cuaternion dual de
    la estructura k.
162          dq2(2:4,:) = -1.*dq2(2:4,:)
163          dq2(6:8,:) = -1.*dq2(6:8,:)
164          DO j=1,m
165              dq1 = dquaternions(l,1:8,j)
166 !!          dq1(2:4) = -1.*dq1(2:4)
167 !!          dq1(6:8) = -1.*dq1(6:8)

```

```

168 !MULTIPLICACION DE LOS CUATERNIONES DE LA ESTRUCTURA 1 POR LOS
      CONJUGADOS
169 !DE LA ESTRUCTURA k. (SE CONSIDERAN SOLO LOS PRODUCTOS INVOLUCRADOS EN
      EL
170 !RESULTADO DE LA PARTE DUAL)
171       CALL Q_PROD_mat(dq1(1:4),dq2(5:8,1:n),pq(1:4,1:n),n)
172
173       CALL Q_PROD_mat(dq1(5:8),dq2(1:4,1:n),qtmp(1:4,1:n),n)
174
175       qresult(1:4,1:n) = pq(1:4,1:n) + qtmp(1:4,1:n)
176 !!En el caso de la medida s se debe aadir la siguiente linea y
      comentar
177 !el resto hasta el termino del ciclo para asi tomar la parte escalar
      del
178 !producto como parametro de comparacion
179 !!       comparison_z(1:n,j) = ABS(qresult(1,1:n))
180
181       qtmp(1:4,1:n) = qresult(1:4,1:n)
182       qtmp(2:4,1:n) = -1.*qtmp(2:4,1:n)
183
184 !Se realiza el producto del cuaternion resultante de la parte dual por
185 !su conjugado (medida z)
186       CALL Q_PROD_mat2(qresult(1:4,1:n),qtmp(1:4,1:n),
187 $               qp(1:4,1:n),n)
188
189       comparison_z(1:n,j) = qp(1,1:n)
190
191 ENDDO
192
193 IF (algorithm==1 .or. algorithm==3) THEN !Algoritmo bmp
194
195       CALL FMINIMUMS(comparison_z(1:n,1:m),minimums(1:min_limit
196 )
197 $               ,n,m,min_limit)
198
199       dm_z(1,k) = SUM(minimums(:min_limit))
200
201       IF (algorithm==1) dm_z(1,k) = dm_z(1,k)/min_limit
202
203 ELSEIF (algorithm == 2) THEN !Algoritmo bmm
204
205       CALL FMINIMUMS_COL(comparison_z(1:n,1:m),
206 $               minimums_col(1:min_limit),n,m,min_limit)
207

```

```

208      CALL FMINIMUMS_ROW( comparison_z(1:n,1:m) ,
209      $      minimums_row(1:min_limit) ,n,m,min_limit)
210
211
212      maxval1=MAXVAL( minimums_col(1:min_limit) )
213      maxval2=MAXVAL( minimums_row(1:min_limit) )
214
215      dm_z(1,k) = MAX(maxval1,maxval2)
216
217      ELSE
218
219      PRINT*, 'ALGORITMO DE BuSQUEDA DE MINIMOS NO ValIDO.
220      $      OPCIONES: (1) BuSQUEDA GLOBAL (bmp) , (2) BuSQUEDA
221      $      POR COLUMNA Y POR RENGLoN (bmm) , (3) bmp SIN
222      $      PROMEDIAR MINIMOS '
223
224      STOP
225
226      ENDIF
227
228      dm_z(k,1) = dm_z(1,k)
229      ENDDO
230      WRITE(101,*) dm_z(1,:)
231      IF (1 == truncate) EXIT
232      ENDDO
233      CLOSE(101)
234
235      ENDIF
236      !TERMINA COMPARACION
237      DEALLOCATE( dquaternions , comparison_z , minimums_col , dm_z ,
238      $      minimums_row , dq2 , qresult , qtmp , pq , qp , minimums ,
239      $      minimums_test , min_loc , mask_search , ref_molecules ,
240      $      T_ref , dqref , mask , structure_size , STAT=estat)
241
242      IF ( estat /=0) PRINT*, 'ERROR DURING DEALLOCATION IN MAIN'
243
244      CALL CPU_TIME( finish )
245
246      WRITE(106,*) 'COMPARISON TIME: ' , finish -start
247      CLOSE(106)
248
249      END PROGRAM

```

```

1      SUBROUTINE GENERATE_MAIN( structure_n , ref_molecule ,
2      $      dquaternions_o , reference_molecule_o )

```

```

3  !      ESTE PROGRAMA CALCULA LOS CUATERNIONES Y CUATERNIONES DUALES DE
      UNA ESTRUCTURA DADA A PARTIR DE LAS COORDENADAS LEIDAS EN EL
      ARCHIVO DE SALIDA DE GENARRIS.

4
5      IMPLICIT NONE
6      INCLUDE 'common.h'
7      INTEGER n, n_aux, n_at, ref, structure_n, ref_molecule
8      CHARACTER line*67, equal*1, line_aux*70
9      REAL err, err_t, coor_ref(3,n_atoms), max_err,
10     $      dquaternions_o(8,size_), reference_molecule_o(3,4),
11     $      reference_molecule(3,4)
12     REAL, ALLOCATABLE :: coor(:,:,:), gcenters(:,:), quaternions(:,:),
13     $      T(:,:), dquaternions_T(:,:)
14     INTEGER, ALLOCATABLE :: labels(:,:)

15
16     n = structure_n
17
18 c      COMIENZA LECTURA DE COORDENADAS EL ARCHIVO DE GENARRIS
19     OPEN(101,FILE='geometry.out')
20
21 10     READ(101,'(A)',ERR=60) line
22
23     IF (line(9:23) == 'BEGIN STRUCTURE') THEN
24         READ(101,*) line, equal, n_aux
25         IF (line=='#structure_number' .AND. n_aux == n) THEN
26             GO TO 20
27         ELSE
28             GO TO 10
29         ENDIF
30     ELSE
31         GO TO 10
32     ENDIF
33
34 60     PRINT*, 'ERROR DURING READING', n
35     STOP
36
37 20     READ(101,'(A)') line
38     READ(line, *,ERR=30) line_aux, equal, n_aux
39
40     IF (line_aux=='#Z') THEN
41         n_molecules = n_aux
42     ENDIF
43
44 30     IF (line(1:14)=='lattice_vector') THEN
45         DO i=1,3

```

```

46         READ(line , *) line_aux , abc(:,i)
47         READ(101,'(A)') line
48     ENDDO
49     GO TO 40
50 ELSE
51     GO TO 20
52 ENDIF
53
54 40 CONTINUE
55
56     n_tot = n_molecules*n_atoms
57
58     zone=start_zone
59     size_ = n_molecules*(zone*2+1)**3-1
60
61     DO WHILE (size_<min_limit) !Determinar el numero de cuaterniones
62 !duales a generar para cada estructura
63         zone=zone+1
64         size_ = n_molecules*(zone*2+1)**3-1
65     ENDDO
66
67     ALLOCATE(coor(n_molecules,3,n_atoms), gcenters(3,n_molecules),
68 $           quaternions(4,size_), labels(4,size_), T(3,size_),
69 $           dquaternions_T(8,size_), STAT=estat)
70
71     IF (estat/=0) THEN
72     PRINT *, 'ERROR DURING ALLOCATION'
73     ENDIF
74
75     n_at=1
76     n_aux=1
77 ! SE ASIGNAN VECTORES DE LA CELDA UNITARIA, SUS COMBINACIONES Y SE
78 ! NORMALIZAN
79     abcdef(1:3,1:3) = abc(1:3,1:3)
80
81     abcdef(1:3,4) = abc(1:3,1)+abc(1:3,2)
82     abcdef(1:3,5) = abc(1:3,1)+abc(1:3,3)
83     abcdef(1:3,6) = abc(1:3,2)+abc(1:3,3)
84     abcdef(1:3,7) = abc(1:3,1)-abc(1:3,2)
85     abcdef(1:3,8) = abc(1:3,1)-abc(1:3,3)
86     abcdef(1:3,9) = abc(1:3,2)-abc(1:3,3)
87     abcdef(1:3,10) = abc(1:3,1)+abc(1:3,2)+abc(1:3,3)
88     abcdef(1:3,11) = abc(1:3,1)-abc(1:3,2)+abc(1:3,3)
89     abcdef(1:3,12) = abc(1:3,1)+abc(1:3,2)-abc(1:3,3)
90     abcdef(1:3,13) = abc(1:3,1)-abc(1:3,2)-abc(1:3,3)

```

```

90
91     DO i=1,13
92         magnitude = SQRT(SUM(abcdef(:,i)**2.))
93         abcdef_abs(i) = magnitude
94         abcdef_u(:,i) = abcdef(:,i) / magnitude
95     ENDDO
96 ! TERMINA VECTORES DE LA CELDA UNITARIA
97 ! LECTURA DE COORDENADAS DE LAS MOLECULAS EN LA CELDA UNITARIA
98     DO i=1,n_tot
99         READ(line,*) line_aux, coor(n_aux,1:3,n_at)
100        n_at = n_at + 1
101        IF (MOD(i,n_atoms)==0) THEN
102            n_aux=n_aux+1
103            n_at = 1
104        ENDIF
105        READ(101,'(A)') line
106    ENDDO
107
108    CLOSE(101)
109 ! TERMINA LECTURA DE COORDENADAS
110 ! SE CALCULAN CENTROS GEOMETRICOS DE CADA MOLECULA EN LA CELDA UNITARIA
111    CALL GEOMETRIC_CENTERS(coor(1:n_molecules,1:3,1:n_atoms),gcenters
112    )
113
114    reference_molecule(1:3,1) = gcenters(1:3,ref_molecule)
115    ref = ref_molecule
116 ! GENERACION DE CUATERNIONES RESPECTO A LA PRIMERA MOLECULA EN LA CELDA
117 ! UNITARIA (ref_molecule=1)
118    CALL GENERATE_QUATERNIONS(coor(1:n_molecules,1:3,1:n_atoms)
119    & ,ref,gcenters,quaternions,labels,T,
120    & reference_molecule(1:3,2:4))
121
122 ! GENERACION DE CUATERNIONES DUALES
123    CALL GENERATE_DUAL_QUATERNIONS(quaternions,T,
124    $ dquaternions_T(:,1:size_),size_)
125
126    IF (algorithm == 2) THEN
127 ! ORDENAMIENTO DE LOS CUATERNIONES DUALES DE MENOR A MAYOR NORMA DE LOS
128 ! VECTORES T CORRESPONDIENTES. SOLO EN EL CASO DEL ALGORITMO bmm
129    CALL ORDER_DUAL_QUATERNIONS(dquaternions_T(1:8,1:size_),
130    $ T(1:3,1:size_),
131    $ dquaternions_o(1:8,1:min_limit))
132    ELSE
133    dquaternions_o(1:8,1:size_) = dquaternions_T(1:8,1:size_)

```

```

134         ENDIF
135
136
137 !VERIFICACION DE CUATERNIONES DUALES (OPCIONAL)
138 !GENERACION DE COORDENADAS RELATIVAS A LA MOLECULA DE REFERENCIA
139 !     do i=1,n_atoms
140 !         coor_ref(1:3,i) = coor(ref,1:3,i) - gcenters(1:3,ref)
141 !     enddo
142 !     CALL VERIFY_DQUAT(coor(:, :, :), coor_ref, dquaternions_T
143 !         $           , n_molecules, n_atoms, labels, gcenters(:, ref))
144 !EN EL ARCHIVO comparison_dq-T.out SE ENCUENTRA EL RMSD ENTRE LAS
145 !COORDENAS
146 !"REALES" Y LAS GENERADAS CON LOS CUATERNIONES DUALES, PARA CADA
147 !MOLECULA.
148 !     OPEN(105, FILE='comparison_dq-T.out', ACTION='READ')
149 !CALCULO DEL ERROR PROMEDIO Y MAXIMO
150 !     err_t = 0.
151 !     max_err = 1.E-20
152 !     DO i=1,size_
153 !         READ(105,*) err
154 !         IF (err > max_err) max_err = err
155 !         err_t = err_t + err
156 !     ENDDO
157 !AQUI SE PUEDE GUARDAR EL ERROR MAXIMO Y PROMEDIO PARA CADA ESTRUCTURA
158 !     OPEN(109, FILE='mean_max_error.out', ACCESS='APPEND', ACTION='WRITE
159 !         ')
160
161 !     WRITE(109,*) err_t/size_, max_err
162
163 !     CLOSE(109)
164
165 !CONDICION ADICIONAL PARA VERIFICAR SI LOS ERRORES SE ENCUENTRAN POR
166 !ENCIMIA DE UN CIERTO VALOR
167 !     IF (max_err > 0.01 .or. err_t /= err_t) then
168 !         PRINT*, "STRUCTURE N", structure_n
169 !         PRINT*, "DUALQUATERNION'S MAX ERROR", max_err
170 !         PRINT*, "MEAN DUALQUATERNION'S ERROR (T)",
171 !         $ err_t/(size_*n_atoms)
172 !     endif
173
174 !     CLOSE(105)
175
176 reference_molecule_o = reference_molecule
177
178 DEALLOCATE(coor, gcenters, quaternions, labels, T, dquaternions_T

```

```

177     $          STAT=estat)
178
179     IF (estat/=0) PRINT *, 'ERROR DURING DEALLOCATION'
180
181     END SUBROUTINE

```

```

1     SUBROUTINE GEOMETRIC.CENTERS(coor_i,gcenters_i)
2     !     ESTA SUBROUTINA GENERA LOS CENTROS GEOMETRICOS DE CADA MOLECULA
3     DADA EN 'coor'.
4     IMPLICIT NONE
5
6     REAL coor_i(n_molecules,3,n_atoms), vgc(3),
7     $      gcenters_i(3,n_molecules), operations_tmp(3,n_tot-1)
8     REAL, ALLOCATABLE :: coor(:,:,:), gcenters(:,:), vtmp(:,:)
9
10    INCLUDE 'common.h'
11
12    ALLOCATE(coor(n_molecules,3,n_atoms), gcenters(3,n_molecules),
13    $      vtmp(3,n_atoms-1), STAT=estat)
14
15    IF (estat /= 0) PRINT *, 'ERROR DURING ALLOCATION
16    $IN GEOMETRIC.CENTERS'
17
18    coor = coor_i
19
20    DO i=1,n_molecules
21        vgc(1:3) = coor(i,1:3,1)
22    !     vgc(1:3) = 0.
23    C     CON LA SUBROUTINA SE VERIFICA LAS COORDENAS DE LOS TOMOS EN UNA
24    MOLCULA SEAN CERCANAS Y NO SEAN SUS IMAGENES PERIODICA
25        CALL FIND-CLOSED(coor(i,1:3,1:n_atoms),vtmp,n_atoms,1
26    $      ,operations_tmp)
27        DO j=1,n_atoms-1
28    !         DO j=1,n_atoms
29            vgc(1:3) = vgc(1:3) + coor(i,1:3,1) + vtmp(1:3,j)
30    !         vgc(1:3) = vgc(1:3) + coor(i,1:3,j)
31    ! Se mueven las coordenadas de los atomos si sus coordenadas no son las
32    mas cercanas al centro geometrico.
33            coor(i,1:3,j+1) = coor(i,1:3,1) + vtmp(1:3,j)
34        ENDDO
35        vgc(1:3) = vgc(1:3)/n_atoms
36        gcenters(1:3,i) = vgc(1:3)
37    ENDDO

```

```

36      gcenters_i = gcenters
37      coor_i = coor
38
39      DEALLOCATE(coor, gcenters, vtmp, STAT=estat)
40
41      IF (estat /= 0) PRINT *, 'ERROR DURING DEALLOCATION
42      $ IN GEOMETRIC.CENTERS'
43
44      ENDSUBROUTINE

```

```

1      SUBROUTINE FIND_CLOSED(coor, vectors_r, n, ref, operations_r)
2      !      ESTA SUBROUTINA ENCUENTRA LOS VECTORES CON MENOR DISTANCIA ENTRE
3      !      TODOS LOS PARES DE COORDENADAS POSIBLES RESPECTO A LA
4      !      COORDENADA DE REFERENCIA (ref) CONSIDERANDO CONDICIONES
5      !      PERIODICAS.
6
7      IMPLICIT NONE
8
9      INTEGER n, vec_i, ref
10     REAL coor(3,n), v(3), operations_r(3,n-1),
11     $      v_abcdef(13), vectors_r(3,n-1), tmp(3), m
12     REAL, ALLOCATABLE :: atoms(:, :), vectors(:, :), operations(:, :)
13
14     INCLUDE 'common.h'
15
16     ALLOCATE(vectors(3,n-1), atoms(3, n_tot),
17     $      operations(3,n-1), STAT=estat)
18
19     IF (estat/=0) THEN
20         PRINT *, 'ERROR DURING ALLOCATION IN FIND_CLOSED'
21     ENDIF
22
23     atoms = coor
24     vec_i=1
25     operations(1:3,1:n-1) = 0.
26     DO i=1,n
27         IF (ref/=i) THEN
28             v = atoms(1:3,i) - atoms(1:3,ref)
29             DO k=1,13
30                 DO l=1,2
31                     m=(-1.0)**l
32                     v_abcdef(k) = DOTPRODUCT(v, m*abcdef_u(1:3,k)) !PROYECCION
33                     DEL VECTOR SOBRE CADA DIRECCION DE LA CELDA UNITARIA Y SUS
34                     DIAGONALES
35                     tmp = REAL(NINT(v_abcdef(k)/abcdef_abs(k)))*m*abcdef(1:3,k)

```

```

32         v = v - tmp
33         operations(1:3,vec_i) = operations(1:3,vec_i) - tmp
34 !   LAS OPERACIONES SOBRE EL VECTOR SE ALMACENAN PARA PODER REALIZAR LAS
35 !   MISMAS OPERACIONES A OTRO ATOMOS SI SE REQUIERE Y ASA NO
36 !   PERDER LA CORRELACION CON EL CENTRO DE MASA.
37         ENDDO
38         ENDDO
39         vectors(1:3,vec_i)=v
40         vec_i=vec_i+1
41     ENDDO
42
43     vectors_r = vectors
44     operations_r = operations
45
46     DEALLOCATE(vectors,atoms,STAT=estat)
47
48     IF (estat/=0) THEN
49         PRINT *, 'ERROR DURING DEALLOCATION IN FIND-CLOSED'
50     ENDDO
51 ENDSUBROUTINE

```

```

1  SUBROUTINE GENERATEQUATERNIONS(coor_i,ref,gcenters_i,
2  $                                quaternions_i,labels_o,
3  $                                translation_o,ref_vecs)
4  !   ESTA SUBROUTINA ENCUENTRA LOS CUATERNIONES DE LA MOLECULA CENTRAL
5  !   CON SUS VECINOS EN LA CELDA UNITARIA Y EN LAS CELDAS UNITARIAS
6  !   VECINAS HASTA EL LIMITE s.
7
8  IMPLICIT NONE
9
10 INCLUDE 'common.h'
11
12 INTEGER di, df, finish, ref, a, m, labels_o(4,size_), step
13 REAL coor_i(n_molecules,3,n_atoms), w(3), coor_ref(3,n_atoms),
14 $    gcenter_ref(3), obases_ref(3,3), ref_vecs(3,3),
15 $    gcenters_i(3,n_molecules), quaternions_i(4,size_),
16 $    coor_ref_rel(3,n_atoms), translation_o(3,size_)
17 REAL, ALLOCATABLE :: coor(:, :, :), coor_tmp(:, :, :),
18 $    obases(:, :, :), quaternions(:, :),
19 $    translation(:, :), gcenters(:, :),
20 $    gcenters_tmp(:, :)
21 INTEGER, ALLOCATABLE :: labels(:, :)
22
23 ALLOCATE(coor(n_molecules,3,n_atoms),
24 $    coor_tmp(n_molecules,3,n_atoms), obases(n_molecules,3,3),

```

```

23      $      quaternions(4,size_),translation(3,size_),
24      $      gcenters(3,n_molecules),gcenters_tmp(3,n_molecules),
25      $      labels(4,size_), STAT=estat)
26
27      IF (estat/=0) PRINT*, 'ERROR DURING ALLOCATION IN PROX'
28
29      coor = coor_i
30      gcenters = gcenters_i
31      coor_ref(1:3,1:n_atoms) = coor_i(ref,1:3,1:n_atoms)
32      gcenter_ref(1:3) = gcenters(1:3,ref)
33
34      DO m=1,n_atoms
35          coor_ref_rel(1:3,m) = coor_ref(1:3,m) - gcenter_ref(1:3)
36      ENDDO
37
38      !SE GENERA UN CONJUNTO ORTONORMAL DE VECTORES PARA LA MOLECULA DE
39      REFERENCIA
40      CALL GENERATEORTONORMALVEC(coor_ref,gcenters(1:3,ref),
41      $      obases_ref,1,n_atoms)
42
43      DO i=1,3
44          ref_vecs(1:3,i) = obases_ref(1:3,i)
45      ENDDO
46
47      di = 1
48      df = di + n_molecules - 1
49      DO i=-zone,zone
50          DO j=-zone,zone
51              DO k=-zone,zone
52                  w(1:3) = i*abc(1:3,1) + j*abc(1:3,2) + k*abc(1:3,3)
53
54      !!SE GENERAN LAS IMAGENES PERIODICAS DE CADA MOLECULA Y DE SUS CENTRO
55      GEOMETRICOS.
56      DO m=1,n_molecules
57          DO a=1,n_atoms
58              coor_tmp(m,1:3,a) = coor(m,1:3,a) + w(1:3)
59          ENDDO
60          gcenters_tmp(1:3,m) = gcenters(1:3,m) + w(1:3)
61      ENDDO
62
63      !
64      IF (i==0 .and. j==0 .and. k==0) THEN
65          finish = n_molecules-2
66          step = 2
67      ELSE
68          finish = n_molecules-1

```

```

66         step = 1
67     ENDIF
68
69     DO m=0,finish
70         translation(1:3,di+m) = gcenters_tmp(1:3,m+step)
71     $                                     -gcenter_ref(1:3)
72         labels(1,di+m) = m+step
73         labels(2,di+m) = i
74         labels(3,di+m) = j
75         labels(4,di+m) = k
76     ENDDO
77
78 !IMPRIME DISTANCIAS RESPECTO A LA MOLECULA DE REFERENCIA
79 !         open(105,file='distances.out')
80 !         do m=0,n_molecules-1
81 !             write(105,*) m+1,i,j,k,
82 !     $             sqrt(sum(translation(di+m,1:3)**2.))
83 !         enddo
84 !         close(105)
85
86 !SE GENERA UN CONJUNTO DE 3 VECTORES ORTONORMALES EQUIVALENTES PARA
87 !CADA MOLECULA
88     CALL GENERATEORTONORMALVEC(coor_tmp,gcenters_tmp,obases,
89     $                             n_molecules,n_atoms)
90
91 !EN LA CELDA UNITARIA CENTRAL SE GENERAN LOS CUATERNIONES DUALES CON
92 !INDICES DIFERETENTES PARA EVITAR CONSIDERAR LA MOLECULA DE REFERENCIA
93     IF (i==0 .and. j==0 .and. k==0) THEN
94         CALL CALCQUATERNIONS(obases(2:n_molecules,1:3,1:3),
95         $             obases_ref,quaternions(1:4,di:df-1),n_molecules-1,
96         $             translation(1:3,di:df-1))
97     !VERIFICAR CUATERNIONES GENERADOS (OPCIONAL)
98     !         CALL VERIFY_QUAT(coor_tmp(2:,:,:),coor_ref_rel,
99     !     $             quaternions(1:4,di:df-1),
100     !     $             gcenters_tmp(1:3,2:n_molecules)
101     !     $             ,n_molecules-1,n_atoms)
102     di = di + n_molecules - 1
103     df = di + n_molecules - 1
104     ELSE
105         CALL CALCQUATERNIONS(obases,obases_ref,
106         $             quaternions(1:4,di:df),n_molecules,
107         $             translation(1:3,di:df))
108     !         CALL VERIFY_QUAT(coor_tmp,coor_ref_rel,
109     !     $             quaternions(1:4,di:df),
110     !     $             gcenters_tmp(1:3,1:n_molecules)

```

```

110 !      $                                ,n_molecules ,n_atoms)
111
112         di = di + n_molecules
113         df = di + n_molecules -1
114     ENDIF
115 ENDDO
116 ENDDO
117 ENDDO
118
119     quaternions_i = quaternions
120     labels_o = labels
121     translation_o = translation
122
123     DEALLOCATE( coor , coor_tmp , obases , quaternions , translation ,
124 $             gcenters , gcenters_tmp , labels , STAT=estat )
125
126     IF ( estat /= 0 ) PRINT* , 'ERROR DURING DEALLOCATION IN PROX'
127
128     ENDSUBROUTINE

```

```

1     SUBROUTINE GENERATEORTONORMALVEC( coor_i , centers , obases_o , n , a )
2 !ESTA SUBROUTINA GENERA UNA TRIADA DE VECTORES ORTONORMALES PARA CADA
3 MOLECULA Y SUS MOLECULAS VECINAS.
4
5     IMPLICIT NONE
6
7     INCLUDE 'common.h'
8     INTEGER n , a
9     REAL coor_i( n , 3 , a ) , obase( 3 , 3 ) , obases( n , 3 , 3 ) ,
10 $      base( 3 , 3 ) , obases_o( n , 3 , 3 ) , centers( 1 : 3 , n )
11     REAL , ALLOCATABLE :: coor( : , : , : )
12
13     ALLOCATE( coor( n , 3 , a ) , STAT=estat )
14
15     IF ( estat /= 0 ) PRINT* , 'ERROR DURING ALLOCATION IN
16 $ GENERATEORTONORMALVEC'
17
18     coor = coor_i
19
20     DO i=1,n
21         base( 1 : 3 , 1 ) = coor( i , 1 : 3 , 1 ) - centers( 1 : 3 , i )
22         base( 1 : 3 , 2 ) = coor( i , 1 : 3 , n_atoms ) - centers( 1 : 3 , i )
23         obase( 1 : 3 , 1 ) = base( 1 : 3 , 1 )
24         obase( 1 : 3 , 2 ) = base( 1 : 3 , 2 ) -
25 $ DOTPRODUCT( base( 1 : 3 , 2 ) , obase( 1 : 3 , 1 ) ) * obase( 1 : 3 , 1 )

```

```

25      $      /SUM(obase(1:3,1)**2.)
26      CALL VPROD(obase(1:3,1),obase(1:3,2),obase(1:3,3))
27
28      DO j=1,3
29          obase(1:3,j) = obase(1:3,j)/SQRT(SUM(obase(1:3,j)**2.))
30      ENDDO
31
32      obases(i,1:3,1:3) = obase(1:3,1:3)
33  ENDDO
34
35      obases_o = obases
36
37      DEALLOCATE(coor,STAT=estat)
38
39      IF (estat/=0) PRINT*, 'ERROR DURING DEALLOCATION IN
40  $ GENERATEORTONORMALVEC'
41
42  ENDSUBROUTINE

```

```

1      SUBROUTINE CALC_QUATERNIONS(coor_i,base_ref,quaternions_i,n,
2      $      translations)
3  !      GENERA LOS CUATERNIONES DE CADA CONJUNTO DE 3 VECTORES
4      ORTONORMALES EN coor_i RESPECTO AL CONJUNTO DE REFERENCIA (base_ref
5      ).
6      IMPLICIT NONE
7
8      INTEGER n
9      REAL coor_i(n,3,3), quaternions_i(4,n), t(3), v1(3),v2(3), theta,
10     $      base_ref(3,3), translations(3,n), rise, v3(3), costheta
11     REAL, ALLOCATABLE :: coor(:,:,:), quaternions(:,:), deltas(:,:,:)
12
13     INCLUDE 'common.h'
14
15     ALLOCATE(coor(n,3,3),quaternions(4,n),deltas(n,3,3),STAT=estat)
16
17     IF (estat/=0) PRINT*, 'ERROR DURING ALLOCATION IN QUATERNIONS'
18
19     coor = coor_i
20
21     DO i=1,n
22         deltas(i,1:3,1:3) = coor(i,1:3,1:3) - base_ref(1:3,1:3)
23         v1 = deltas(i,1:3,3)-deltas(i,1:3,1)
24         v2 = deltas(i,1:3,3)-deltas(i,1:3,2)
25         CALL VPROD(v1,v2,t)
26         IF (SQRT(SUM(v1**2.))>1.E-5 .AND. SQRT(SUM(v2**2.))>1.E-5)

```

```

25     $ THEN
26         t = t/SQRT(SUM(t**2.))
27         v1 = base_ref(1:3,1) -
28     &         dot_product(base_ref(1:3,1),t)*t
29         v2 = coor(i,1:3,1) -
30     &         dot_product(coor(i,1:3,1),t)*t
31         v1 = v1/SQRT(SUM(v1**2.))
32         v2 = v2/SQRT(SUM(v2**2.))
33     ELSE
34         t = 0.0
35         v1 = 0.0
36         v2 = 0.0
37         v1(1) = 1.0
38         v2(1) = 1.0
39     ENDIF
40     costheta = DOTPRODUCT(v1,v2)
41 1     theta = ACOS(costheta)
42 !ESTA CONDICION EVITAR INDETERMINACIONES POR ERROR NUMERICO CUANDO
43 !costheta ES LIGERAMENTE SUPERIOR A 1 O LIGERAMENTE INFERIOR A -1
44     IF (theta /= theta) THEN
45         costheta=FLOAT(INT(costheta))
46         GO TO 1
47     ENDIF
48     rise = DOTPRODUCT(translations(:,i),t)
49     IF (rise < 0.) THEN
50 !         rise(i) = -1.*rise(i)
51         t = -1.*t
52     ENDIF
53     CALL VPROD(v1,v2,v3)
54     IF (DOTPRODUCT(v3,t) < 0.) theta = 2.*pi - theta
55     quaternions(1,i) = COS(theta/2.)
56     quaternions(2:4,i) = SIN(theta/2.)*t(1:3)
57 ENDDO
58
59     quaternions_i = quaternions
60
61     DEALLOCATE(coor,quaternions,deltas,STAT=estat)
62
63     IF (estat/=0) PRINT*, 'ERROR DURING DEALLOCATION IN QUATERNIONS'
64 ENDSUBROUTINE

```

```

1 SUBROUTINE GENERATEDUALQUATERNIONS(quaternions_i,T_i,
2 $                                     dquaternions_T_o,n)
3 ! GENERA LOS CUATERNIONES DUALES A PARTIR DE LOS CUATERNIONES Y
4 ! VECTORES T ENTRE CADA MOLECULA Y LA MOLECULA DE REFERENCIA.

```

```

5      IMPLICIT NONE
6
7      INTEGER n
8      REAL quaternions_i(4,n) , T_i(3,n) , dquaternions_T_o(8,n) , t_tmp
(4)
9      $      , qt(4)
10     REAL, ALLOCATABLE :: quaternions(:, :) , dquaternions_T(:, :) , T
(:, :)
11
12     INCLUDE 'common.h'
13
14     ALLOCATE(quaternions(4,n) , dquaternions_T(8,n) , T(3,n)
15     $      , STAT=estat)
16
17     IF (estat /= 0) PRINT*, 'ERROR DURING ALLOCATION
18     $      IN DUALQUATERNIONS'
19
20     quaternions = quaternions_i
21     T = T_i
22
23     DO i=1,n
24         dquaternions_T(1:4,i) = quaternions(1:4,i)
25         t_tmp(1) = 0.
26     !AQUI SE UTILIZA EL VECTOR ENTRE CENTROS GEOMETRICOS
27         t_tmp(2:4) = T(1:3,i)
28         CALL QPROD(t_tmp, quaternions(1:4,i) , qt)
29         qt(1:4) = 0.5*qt(1:4)
30         dquaternions_T(5:8,i) = qt(1:4)
31     ENDDO
32
33     dquaternions_T_o(1:8,1:n) = dquaternions_T(1:8,1:n)
34
35     DEALLOCATE(quaternions , T, dquaternions_T , STAT=estat)
36
37     IF (estat /= 0) PRINT*, 'ERROR DURING DEALLOCATION
38     $      IN DUALQUATERNIONS'
39
40     ENDSUBROUTINE

```

```

1      SUBROUTINE ORDER_DUALQUATERNIONS(dq,T,dq_o)
2      !ESTA SUBROUTINA ORDENAR LOS CUATERNIONES DUALES DEL CONJUNTO dq DE
3      !MENOR A MAYOR (HASTA EL LIMITE min_limit) EN FUNCION DE LA NORMA DE
4      LOS VECTOR EN T
5      IMPLICIT NONE
6      INCLUDE 'common.h'

```

```

6
7      REAL, ALLOCATABLE :: dq_ordered(:, :), distance(:)
8      LOGICAL, ALLOCATABLE :: mask(:)
9      REAL dq(8, size_), T(3, size_), dq_o(8, min_limit)
10     INTEGER loc(1)
11
12     ALLOCATE(dq_ordered(8, min_limit), distance(size_), mask(size_))
13     $          , STAT=estat)
14
15     IF (estat /= 0) PRINT*, 'ERROR DURING ALLOCATION IN ORDER
16     DUALQUATERN
17     $IONS '
18
19     !CALCULO DE LA NORMA DE LOS VECTORES T
20     distance(1: size_) = SQRT(SUM(T(1:3, 1: size_) * T(1:3, 1: size_), DIM=1))
21
22     !ORDENAMIENTO DE MENOR A MAYOR HASTA EL LIMITE min_limit
23     mask = .TRUE.
24     DO i = 1, min_limit
25         loc = MINLOC(distance(1: size_), MASK=mask(1: size_))
26         dq_ordered(1:8, i) = dq(1:8, loc(1))
27         mask(loc(1)) = .FALSE.
28     ENDDO
29
30     dq_o(1:8, 1: min_limit) = dq_ordered(1:8, 1: min_limit)
31
32     DEALLOCATE(dq_ordered, distance, mask, STAT=estat)
33
34     IF (estat /= 0) PRINT*, 'ERROR DURING DEALLOCATION
35     $          IN ORDER DUALQUATERNIONS'
36
37     ENDSUBROUTINE

```

```

1      SUBROUTINE QPROD(p, q, r)
2      !      ESTA SUBROUTINA RALIZA EL PRODUCTO DE DOS CUATERNIONES.
3      IMPLICIT NONE
4      REAL, DIMENSION (4) :: p, q, r, tmp
5
6      r(1) = p(1)*q(1) - DOT_PRODUCT(p(2:4), q(2:4))
7      r(2:4) = p(1)*q(2:4) + q(1)*p(2:4)
8      CALL VPROD(p(2:4), q(2:4), tmp(2:4))
9      r(2:4) = r(2:4) + tmp(2:4)
10
11     ENDSUBROUTINE

```

```

1      SUBROUTINE Q_PROD_mat(s1 , s2 , sr , n)
2      !REALIZA EL PRODUCTO DE UN CUATERNION CON UNA MATRIZ DE CUATERNIONES
3      IMPLICIT NONE
4
5      REAL s1 (4) , s2 (4 , n) , s2_t (n , 4) , sr (4 , n) , sr_cross (3 , n) , saux (3 , n)
6      INTEGER n , i
7
8      s2_t = TRANSPOSE(s2)
9
10     sr (1 , 1:n) = s1 (1) * s2 (1 , 1:n) - MATMUL(s1 (2:4) , s2 (2:4 , 1:n))
11
12     DO i=1,3
13         saux(i , 1:n) = s1 (i+1) * s2 (1 , 1:n)
14     ENDDO
15
16     sr (2:4 , 1:n) = s1 (1) * s2 (2:4 , 1:n) + saux (1:3 , 1:n)
17
18     sr_cross (1 , 1:n) = s1 (3) * s2 (4 , 1:n) - s1 (4) * s2 (3 , 1:n)
19     sr_cross (2 , 1:n) = s1 (4) * s2 (2 , 1:n) - s1 (2) * s2 (4 , 1:n)
20     sr_cross (3 , 1:n) = s1 (2) * s2 (3 , 1:n) - s1 (3) * s2 (2 , 1:n)
21
22     sr (2:4 , 1:n) = sr (2:4 , 1:n) + sr_cross (1:3 , 1:n)
23
24     ENDSUBROUTINE

```

```

1      SUBROUTINE Q_PROD_mat2(s1 , s2 , sr , n)
2      !REALIZA EL PRODUCTO DE UNA MATRIZ DE CUATERNIONES CON UNA MATRIZ DE
3      CUATERNIONES
4      IMPLICIT NONE
5
6      REAL s1 (4 , n) , s2 (4 , n) , s2_t (n , 4) , sr (4 , n) , sr_cross (3 , n) , saux (3 , n)
7      INTEGER n , i
8
9      s2_t = TRANSPOSE(s2)
10
11     sr (1 , 1:n) = s1 (1 , 1:n) * s2 (1 , 1:n) -
12     $          SUM(s2 (2:4 , 1:n) * s1 (2:4 , 1:n) , DIM=1)
13
14     DO i=1,3
15         saux(i , 1:n) = s1 (1 , 1:n) * s2 (i+1 , 1:n) + s2 (1 , 1:n) * s1 (i+1 , 1:n)
16     ENDDO
17
18     sr_cross (1 , 1:n) = s1 (3 , 1:n) * s2 (4 , 1:n) - s1 (4 , 1:n) * s2 (3 , 1:n)
19     sr_cross (2 , 1:n) = s1 (4 , 1:n) * s2 (2 , 1:n) - s1 (2 , 1:n) * s2 (4 , 1:n)
20     sr_cross (3 , 1:n) = s1 (2 , 1:n) * s2 (3 , 1:n) - s1 (3 , 1:n) * s2 (2 , 1:n)

```

```

20
21      sr(2:4,1:n) = saux(1:3,1:n) + sr_cross(1:3,1:n)
22
23      ENDSUBROUTINE

```

```

1      SUBROUTINE FMINIMUMS(d_i, minimums_o, n, m, limit)
2      !ALGORITMO DE BUSQUEDA DE MINIMOS GLOBALES EN MATRICES O ALGORITMO DE
3      !BUSQUEDA DEL MINIMO PROMEDIO (bmp)
4      IMPLICIT NONE
5
6      INTEGER i, estat, limit, n, m, loc(2)
7      REAL d_i(n,m), minimums_o(limit)
8      REAL, ALLOCATABLE :: d(:, :), minimums(:)
9
10     ALLOCATE(d(n,m), minimums(limit), STAT=estat)
11
12     IF (estat /= 0) PRINT*, 'ERROR DURING ALLOCATION IN MINIMUMS'
13
14     d = d_i
15
16     DO i=1, limit
17         loc = MINLOC(d) !BUSQUEDA GLOBAL DE UN MINIMO EN LA MATRIZ d
18         minimums(i) = d(loc(1), loc(2))
19     !SE ANULA LA CONSIDERACION DEL RENGLON Y LA COLUMNA DEL MINIMO
20     !ENCONTRADO PARA LAS BUSQUEDAS SUBSECUENTES.
21         d(loc(1), :) = 1E30
22         d(:, loc(2)) = 1E30
23     ENDDO
24
25     minimums_o = minimums
26
27     DEALLOCATE(d, minimums, STAT=estat)
28
29     IF (estat /= 0) PRINT*, 'ERROR DURING DEALLOCATION IN MINIMUMS'
30     ENDSUBROUTINE

```

```

1      SUBROUTINE FMINIMUMS_ROW(d_i, minimums_o, n, m, limit)
2      !BUSQUEDA DE MINIMOS POR RENGLON DE UNA MATRIZ CUADRADA
3      IMPLICIT NONE
4
5      INTEGER i, j, jc, jo, estat, n, loc(1), m, s, s2, limit
6      REAL d_i(n,m), minimums_o(limit), v1, v2
7      REAL, ALLOCATABLE :: d(:, :), minimums(:)
8      INTEGER, ALLOCATABLE :: pos(:)
9      LOGICAL test
10     LOGICAL, ALLOCATABLE :: avail(:, :)

```

```

11
12     s = MIN(n,m)
13     s2 = MAX(n,m)
14
15     ALLOCATE(d(n,m),minimums(s),avail(n,m),pos(s2),STAT=estat)
16
17     IF (estat/=0) PRINT*, 'ERROR DURING ALLOCATION IN MINIMUMS'
18
19     d = d_i
20
21     avail = .true.
22     pos = 0
23     DO j=1,s
24         jc = j
25         test=.true.
26         DO WHILE (test)
27             loc = MINLOC(d(jc,:), MASK = avail(jc,:))
28             v2 = d(jc,loc(1))
29             avail(jc,loc(1)) = .FALSE.
30             IF(pos(loc(1)).eq.0) THEN !Se conserva el minimo si no se ha
31 !encontrado otro minimo en la misma columna
32                 minimums(jc) = v2
33                 pos(loc(1)) = jc !Se guarda la columna en la que se encontro
34 !el minimo en la posicion del renglon asociado
35                 test = .FALSE.
36             ELSE
37                 v1 = d(pos(loc(1)),loc(1)) !Minimo previo con la misma
columna
38                 IF (v2.gt.v1) THEN !Se conserva el valor mas grande
39                     minimums(jc)=v2
40                     jo = pos(loc(1))
41                     pos(loc(1)) = jc
42                     jc = jo !Se modifica para repetir la busqueda en el renglon
43 !del minimo descartado
44                 END IF
45             END IF
46         END DO
47     END DO
48
49     minimums_o = minimums
50
51     DEALLOCATE(d,minimums,avail,pos,STAT=estat)
52
53     IF (estat/=0) PRINT*, 'ERROR DURING DEALLOCATION IN MINIMUMS'
54

```

11.1.2. Cálculo de la medida *RMSD*

Para el algoritmo de la medida *RMSD* se trabajó sobre el mismo programa de la medida *z*, mostrado en la sección anterior. Por lo tanto, solo se incluyen las líneas que difieren respecto al algoritmo anterior y una subrutina adicional.

```

1  !Lineas 1 a 146 del PROGRAMA COMPARE STRUCTURES Z
2  !COMIENZA COMPARACION DE ESTRUCTURAS
3      WRITE(102,*) 'DUAL QUATERNIONS TIME: ', t
4      DO l=1,n_structures
5          n = structure_size(l)
6          DO k=1,n_structures
7              m = structure_size(k)
8              me = 0.
9  !TRANSFORMACION DE LOS CUATERNIONES DE LA ESTRUCTURA k A LA REFERENCIA
   !DE LA ESTRUCTURA l
10         dq2(1:8,1:m) = dquaternions(k,1:8,1:m)
11 !SE CAMBIA POR TRANSFORMACION DE k A l
12 !         dqref(1,k,2:4) = -1.*dqref(1,k,2:4)
13 !         dqref(1,k,6:8) = -1.*dqref(1,k,6:8)
14 !-----
15 !dqrefcj: SEGUNDO CONJUGADO DEL CUATERNIN DUAL ENTRE REFERENCIAS
16         dqrefcj(1:8) = dqref(1,1:8,k)
17         dqrefcj(2:4) = -1.*dqrefcj(2:4)
18         dqrefcj(6:8) = -1.*dqrefcj(6:8)
19         DO i=1,m
20             CALL DQ_PROD(dq2(1:8,i),dqref(1,1:8,k),dq_tmp(1:8))
21             CALL DQ_PROD(dqrefcj(1:8),dq_tmp(1:8),dq2(1:8,i))
22         ENDDO
23
24
25 !TERMINA TRANSFORMACION
26 !Aquí se utiliza el segundo conjugado del segundo cuaternin dual de
   !la estructura k.
27         dq2cj(1:8,1:m) = dq2(1:8,1:m)
28         dq2cj(2:4,1:m) = -1.*dq2cj(2:4,1:m)
29         dq2cj(5,1:m) = -1.*dq2cj(5,1:m)
30         DO j=1,n
31             dq1 = dquaternions(1,1:8,j)
32
33 !SE GENERAN LAS COORDENADAS DE LOS TOMOS DE LA ESTRUCTURA l POR CADA
   !CUATERNIN DUAL j
34         CALL APPLY_DQ(ref_atoms(1,1:3,1:n_atoms),

```

```

35      $                                coor1(1:3,1:n_atoms),dq1,n_atoms)
36
37
38 !SE APLICA EL CUATERNIN q de k sobre la mol cula de referencia de l
   (m1) EN LA ESTRUCTURA l Y SE COMPARAN LAS COORDENAS RESULTANTES
39      m1(1:8) = 0.
40      m1(1) = 1.
41
42      DO i=1,n_atoms
43          m1(6:8) = ref_atoms(l,1:3,i)
44 !
45          CALL DQ_PROD_mat(m1,dq2cj(1:8,1:m),mltmp(1:8,1:m),
46      $                                m)
47 !
48          CALL DQ_PROD_mat2(dq2(1:8,1:m),mltmp(1:8,1:m),
49      $                                m1dq(1:8,1:m),m)
50
51 !SE MIDE EL RMSD ENTRE LAS COORDENADAS RESULTANTES m1dq Y LAS "REALES"
52      DO a=1,m
53          me(j,a) = me(j,a)+SUM((m1dq(6:8,a)-
54      $                                coor1(1:3,i))*2.)
55      ENDDO
56      ENDDO
57      me(j,1:m)=SQRT(me(j,1:m)/n_atoms)
58
59      ENDDO
60 !Lineas 195 a 259 del PROGRAMA COMPARE STRUCTURES Z,
61 !tal que me=comparison_z

```

```

1      SUBROUTINE APPLY_DQ(coor,coor_r,dq,n)
2 !ESTA SUBROUTINA APLICA EL CUATERNIONES DUAL dq A CADA COORDENADA EN
   coor Y REGRESA LAS COORDENADAS RESULTANTES EN coor_r
3
4      IMPLICIT NONE
5      INTEGER n, i
6      REAL coor(3,n), coor_r(3,n), dq(8),
7      $      q(8), q_(8), r(8), tmpq(8), v(3), vtmp(3), tmpq_(8) ,
8      $      gcenter_ref(3)
9
10     q_(:) = dq(1:8)
11     q_(2:4) = -1.*q_(2:4)
12     q_(5) = -1.*q_(5)
13     DO i=1,n
14         tmpq(:) = 0.
15         tmpq(1) = 1.

```

```

16      tmpq(6:8) = coor(1:3,i)
17
18      CALL DQ_PROD(tmpq,q-,r)
19
20      CALL DQ_PROD(dq(1:8),r,tmpq-)
21
22      coor_r(1:3,i) = tmpq_(6:8)
23
24      ENDDO
25
26      ENDSUBROUTINE

```

11.1.3. Cálculo de la medida *RCD*

De manera similar a la sección anterior, en esta solo se muestran aquellas partes que difieren de los códigos previamente incluidos.

```

1      PROGRAM COMPARE STRUCTURES RCD
2
3      IMPLICIT NONE
4
5      REAL, ALLOCATABLE :: rcd(:,:,:),D(:,:),minimums(:),
6      $                  distance_matrix(:,:), v(:,:), v_(:),
7      $                  minimums_row(:), minimums_col(:)
8      CHARACTER comment, msg*100, auxc*50, dm_name*50, t_name*50
9      REAL diff_T, diff_Q, p1(3), p2(3), q1(3), q2(3), k-, x-,y-,
10     $ start, finish, maxval1, maxval2
11     INTEGER n_structures, count-, total_dq, step,
12     $ ref_molecule, initial_struct, final_struct, partition,
13     $ max_size_l(30), n_molecules_s, n_molecules_l(30),
14     $ n, m, truncate, rcd_limit
15     INTEGER, ALLOCATABLE :: structure_size(:)
16     INCLUDE 'common.h'
17
18     OPEN(101,FILE='input')
19
20     READ(101,*) comment, initial_struct, final_struct
21     READ(101,*) comment, n_molecules_s
22     DO i=1,n_molecules_s
23         READ(101,*) n_molecules_l(i)
24     ENDDO
25     READ(101,*) comment, n_atoms
26     READ(101,*) comment, start_zone
27     READ(101,*) comment, truncate
28     READ(101,*) comment, min_limit

```

```

29      READ(101,*) comment, algorithm
30
31      CLOSE(101)
32
33      n_structures = final_struct - initial_struct + 1
34      size_ = n_molecules*(zone*2+1)**3-1
35
36      IF (min_limit == 0) min_limit = size_ !Cantidad de minimos a
sumar para cada matriz D
37
38      n_structures = final_struct-initial_struct + 1
39
40      DO i=1,n_molecules_s
41          zone = start_zone
42          max_size = n_molecules_l(i)*(zone*2+1)**3-1
43          DO WHILE (max_size<min_limit)
44              zone = zone+1
45              max_size = n_molecules_l(i)*(zone*2+1)**3-1
46          ENDDO
47          max_size_l(i) = max_size
48      ENDDO
49
50      max_size = MAXVAL(max_size_l(:n_molecules_s))
51
52      ALLOCATE(rcd(n_structures,6,max_size),
53      $      D(max_size,max_size),
54      $      minimums(min_limit),
55      $      distance_matrix(n_structures,n_structures),
56      $      v(3,max_size),v_(max_size),
57      $      structure_size(n_structures),
58      $      minimums_col(max_size),
59      $      minimums_row(max_size),
60      $      STAT=estat,errmsg=msg)
61
62      IF (estat/=0) PRINT*, 'ERROR DURING ALLOCATION IN MAIN', estat,
msg
63
64      ref_molecule = 1
65
66      CALL CPU_TIME(start)
67      OPEN(102,FILE='list.out')
68
69      count_=1
70      IF (algorithm == 2) THEN
71          rcd_limit = min_limit

```

```

72     structure_size = min_limit
73 ELSE
74     rcd_limit = max_size
75 ENDIF
76 DO i=initial_struct , final_struct
77 !SE GENERA EL CONJUNTO DE VECTORES RCD PARA CADA ESTRUCTURA
78     CALL GENERATEMAIN(i,1,rcd(count_,1:6,1:rcd_limit))
79     IF (algorithm /= 2) structure_size(count_) = size_
80     count_=count_+1
81 ENDDO
82
83 CLOSE(102)
84 CALL CPU_TIME(finish)
85
86
87 !FORMATO DEL ARCHIVO DE SALIDA
88     IF (n_structures > 1) THEN
89 WRITE(auxc,'(I8)') initial_struct
90 dm_name = 'dm_rcd_'//TRIM(ADJUSTL(auxc))//'_-'
91 t_name = 'dm_rcd_'//TRIM(ADJUSTL(auxc))//'_-'
92 IF (truncate /= 0) THEN
93     WRITE(auxc,'(I8)') truncate
94 ELSE
95     WRITE(auxc,'(I8)') final_struct
96 ENDIF
97 dm_name = TRIM(dm_name)//TRIM(ADJUSTL(auxc))//'.out'
98 t_name = TRIM(t_name)//TRIM(ADJUSTL(auxc))//'_time.out'
99 OPEN(105,FILE=t_name)
100 WRITE(105,*) 'GEOM TIME: ', finish-start
101 CALL CPU_TIME(start)
102 OPEN(102,FILE=dm_name)
103 count_ = 1
104 k_ = 1.
105 !INICIO DE COMPARACION DE ESTRUCTURAS
106 DO l=1,n_structures
107     n = structure_size(l)
108     DO k=1,n_structures
109         m = structure_size(k)
110         DO i=1,n
111             DO j=1,m
112 !GENERACION DE MATRIZ D, DIFERENCIA ENTRE PARES DE VECTORES PARA CADA
113             PAR DE MOLECULAS, POR CADA PAR DE ESTRUCTURAS.
114                 p1 = rcd(l,1:3,i)
115                 p2 = rcd(k,1:3,j)
116                 q1 = rcd(l,4:6,i)

```

```

116         q2 = rcd(k,4:6,j)
117         diff_T = SQRT(SUM(p1**2.))*SQRT(SUM(p2**2.))
118         diff_T = SUM((p1-p2)**2.)/diff_T
119         diff_Q = k_*SUM((q1-q2)**2.)/3.
120
121         D(i,j) = diff_T + diff_Q
122
123     ENDDO
124 ENDDO
125 !SE GENERA MATRIZ DE DISTANCIA
126 IF (algorithm==1) THEN
127
128     CALL FMINIMUMS(D(1:n,1:m),minimums(1:min_limit)
129 $           ,n,m,min_limit)
130
131     distance_matrix(l,k) = SUM(minimums(:min_limit))/
min_limit
132
133 ELSEIF (algorithm == 2) THEN
134     CALL FMINIMUMS_COL(D(1:n,1:m),
135 $           minimums_col(1:min_limit),n,m,min_limit)
136
137
138     CALL FMINIMUMS_ROW(D(1:n,1:m),
139 $           minimums_row(1:min_limit),n,m,min_limit)
140
141
142     IF (min_limit == MIN(n,m)) THEN
143
144         maxval1=MAXVAL(minimums_col(1:min_limit))
145         maxval2=MAXVAL(minimums_row(1:min_limit))
146
147
148     ELSE
149
150         maxval1=minimums_col(min_limit)
151         maxval2=minimums_row(min_limit)
152
153     ENDF
154
155     distance_matrix(l,k) = MAX(maxval1,maxval2)
156
157 ELSEIF (algorithm == 3) THEN
158
159     CALL FMINIMUMS(D(1:n,1:m),minimums(1:min_limit),

```

```

160      $                                n,m,min_limit)
161
162      distance_matrix(l,k) = SUM(minimums(1:min_limit))
163
164      ELSE
165
166      PRINT*, 'ALGORITMO DE BSQUEDA DE MNIMOS NO V LIDO .
167      $          OPCIONES: (1) BSQUEDA GLOBAL, (2) BSQUEDA
168      $          POR COLUMNA'
169
170      ENDIF
171
172      distance_matrix(k,l) = distance_matrix(l,k)
173      count_ = count_ + 1
174      ENDDO
175      WRITE(102,*) distance_matrix(l,:)
176      IF (l == truncate) EXIT
177      ENDDO
178      CLOSE(102)
179
180      ENDIF
181
182      DEALLOCATE(rcd,STAT=estat)
183
184      IF (estat/=0) PRINT*, 'ERROR DURING DEALLOCATION IN MAIN'
185
186      CALL CPU_TIME(finish)
187
188      WRITE(105,*) 'COMPARISON TIME:', finish-start
189
190      CLOSE(105)
191
192      END PROGRAM

```

```

1  !Lineas 1 a 111 de la SUBROUTINA GENERATEMAIN
2  !SE GENERA EL CONJUNTO DE VECTORES RCD PARA LA ESTRUCTURA DADA
3      CALL GENERATEPQ(coor(1:n_molecules,1:3,1:n_atoms)
4      &                ,ref,gcenters,labels,rcd)
5
6      IF (algorithm == 2) THEN
7          CALL ORDERDUALQUATERNIONS(rcd(1:6,1:size_),
8          $                                rcd(1:3,1:size_),
9          $                                rcd_o(1:6,1:min_limit))
10         ELSE
11             rcd_o(1:6,1:size_) = rcd(1:6,1:size_)

```

```

12      ENDIF
13
14      DEALLOCATE(coor , gcenters , labels , T, rel_orientations , STAT=estat)
15
16      IF ( estat /= 0) PRINT *, 'ERROR DURING DEALLOCATION'
17
18      END SUBROUTINE

```

```

1      SUBROUTINE GENERATEPQ(coor_i , ref , gcenters_i ,
2          $          labels_o , rcd_o)
3      !      ESTA SUBROUTINA ENCUENTRA LOS PARES DE VECTORES RCD ENTRE CADA
4      !      MOLCULA EN EL CRISTAL Y LA MOLCULA DE REFERENCIA.
5
6      IMPLICIT NONE
7
8      INCLUDE 'common.h'
9
10     INTEGER n, di , count_ , index_ , n_mol , df ,
11     $      ref , nt , a , m , labels_o(4,size_) , initialmol
12     REAL coor_i(n_molecules , 3 , n_atoms) , w(3) , coor_ref(3,n_atoms) ,
13     $      gcenter_ref(3) , obases_ref(3,3) , gcenters_i(3,n_molecules) ,
14     $      coor_ref_rel(3,n_atoms) ,
15     $      rcd_o(6,size_)
16     REAL, ALLOCATABLE :: distances (:,:) , coor (:,:,) , coor_tmp (:,:,)
17     ,
18     $      obases (:,:,) , translation (:,:) , gcenters (:,:)
19     ,
20     $      gcenters_tmp (:,:) , rcd (:,:)
21     INTEGER, ALLOCATABLE :: labels (:,:)
22
23     ALLOCATE(coor(n_molecules , 3 , n_atoms) ,
24     $      coor_tmp(n_molecules , 3 , n_atoms) , obases(n_molecules , 3 , 3) ,
25     $      gcenters(3,n_molecules) , gcenters_tmp(3,n_molecules) ,
26     $      labels(4,max_size) ,
27     $      rcd(6,max_size) ,
28     $      STAT=estat)
29
30     IF ( estat /= 0) PRINT*, 'ERROR DURING ALLOCATION IN PROX'
31
32
33     coor = coor_i
34     gcenters = gcenters_i
35     coor_ref(1:3,1:n_atoms) = coor_i(ref,1:3,1:n_atoms)
36     gcenter_ref(1:3) = gcenters(1:3,ref)
37
38     DO m=1,n_atoms

```

```

35      coor_ref_rel(1:3,m) = coor_ref(1:3,m) - gcenter_ref(1:3)
36      ENDDO
37
38 !SE GENERA UN CONJUNTO ORTONORMAL DE VECTORES PARA LA MOLCULA DE
REFERENCIA
39      CALL GENERATEORTONORMALVEC( coor_ref , gcenters(1:3,ref) ,
40      $      obases_ref , 1 , n_atoms)
41
42      di = 1
43      df=n_molecules
44      DO i=-zone , zone
45          DO j=-zone , zone
46              DO k=-zone , zone
47                  w(1:3) = i*abc(1:3,1) + j*abc(1:3,2) + k*abc(1:3,3)
48
49 !SE GENERAN LAS IMAGENES PERIODICAS DE CADA MOLCULA Y DE SUS CENTRO
GEOMTRICOS.
50          DO m=1,n_molecules
51              DO a=1,n_atoms
52                  coor_tmp(m,1:3,a) = coor(m,1:3,a) + w(1:3)
53              ENDDO
54              gcenters_tmp(1:3,m) = gcenters(1:3,m) + w(1:3)
55          ENDDO
56
57 !SE GENERA UN CONJUNTO DE 3 VECTORES ORTONORMALES EQUIVALENTES PARA
CADA MOLCULA
58
59      CALL GENERATEORTONORMALVEC( coor_tmp , gcenters_tmp , obases ,
60      $      n_molecules , n_atoms)
61
62 !Condiciones para evitar comparar la molecula de referencia consigo
misma.
63      IF ( i==0 .AND. j==0 .AND. k==0) THEN
64          initialmol = 1
65          di = di-1
66          df = df-1
67      ELSE
68          initialmol = 0
69      ENDIF
70      DO m=initialmol , n_molecules-1
71 !SE OBTIENEN LOS VECTORES ENTRE CENTROS GEOMTRICOS
72          rcd(1:3,di+m) = gcenters_tmp(1:3,m+1)
73      $      -gcenter_ref(1:3)
74 !SE OBTIENEN LAS ORIENTACIONES RELATIVAS
75      DO a=1,3

```

```

76         rcd(3+a,di+m) =
77     $         DOT_PRODUCT(obases(m+1,1:3,a),obases_ref(1:3,a))
78     ENDDO
79         labels(1,di+m) = m+1
80         labels(2,di+m) = i
81         labels(3,di+m) = j
82         labels(4,di+m) = k
83     ENDDO
84         di = di + n_molecules
85         df = df + n_molecules
86     ENDDO
87 ENDDO
88 ENDDO
89
90 labels_o = labels
91 rcd_o = rcd
92
93 DEALLOCATE(coor,coor_tmp,obases,gcenters,gcenters_tmp,
94 $         labels,rcd,STAT=estat)
95
96 IF (estat /= 0) PRINT*, 'ERROR DURING DEALLOCATION IN PROX'
97
98 ENDSUBROUTINE

```



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE EXAMEN DE GRADO

No. 00114

Matrícula: 2211800549

Análisis configuracional de
cristales moleculares
empleando cuaterniones duales

En la Ciudad de México, se presentaron a las 11:30 horas
del día 24 del mes de mayo del año 2023 en la Unidad
Iztapalapa de la Universidad Autónoma Metropolitana, los
suscritos miembros del jurado:

DR. HERBERT HÖPFL BACHNER
DR. MARCELO ENRIQUE GALVAN ESPINOSA
DR. RODOLFO OCTAVIO ESQUIVEL OLEA



HAZEL URIEL VAZQUEZ HERNANDEZ
ALUMNO

Bajo la Presidencia del primero y con carácter de
Secretario el último, se reunieron para proceder al Examen
de Grado cuya denominación aparece al margen, para la
obtención del grado de:

MAESTRO EN CIENCIAS (QUIMICA)

DE: HAZEL URIEL VAZQUEZ HERNANDEZ

y de acuerdo con el artículo 78 fracción III del
Reglamento de Estudios Superiores de la Universidad
Autónoma Metropolitana, los miembros del jurado
resolvieron:

— APROBAR —

REVISÓ

MTRA. ROSALÍA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

Acto continuo, el presidente del jurado comunicó al
interesado el resultado de la evaluación y, en caso
aprobatorio, le fue tomada la protesta.

DIRECTOR DE LA DIVISIÓN DE CBI

DR. ROMAN LINARES ROMERO

PRESIDENTE

DR. HERBERT HÖPFL BACHNER

VOCAL

DR. MARCELO ENRIQUE GALVAN ESPINOSA

SECRETARIO

DR. RODOLFO OCTAVIO ESQUIVEL OLEA

