



MASTERARBEIT | MASTER'S THESIS

Titel | Title

Characterization of a putative cholesterol binding pocket in the Kv10.1 potassium channel

verfasst von | submitted by

Amil Kakkattu BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Magister pharmaciae (Mag. pharm.)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt |
Degree programme code as it appears on the
student record sheet:

UA 066 605

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Pharmazie

Betreut von | Supervisor:

Assoz. Prof. Mag. Dr. Anna Weininger Privatdoz.

Abstract

The Kv10.1 potassium channel is a voltage-gated ion channel, which is widely overexpressed in more than 70% of human tumors, where it has been associated with cell proliferation, angiogenesis and tumor cell migration. Despite its relevance as a pharmacological target, the structural basis of its regulation by membrane lipids remains poorly understood. Computational docking studies performed on the cryo-EM structure of Kv10.1 (PDB: 5K7L) suggested the existence of a potential cholesterol interaction region. However, whether this region represents a stable and physiologically relevant binding pocket under realistic membrane conditions has not yet been established.

This thesis addresses these questions by analyzing coarse-grained molecular dynamics (MD) simulations of the Kv10.1 tetramer embedded in a cholesterol-containing lipid bilayer using the Martini force field. Two independent simulation runs of approximately 1.5 μ s each were analyzed, focusing on a set of eight residues.

Two-dimensional cholesterol density maps of the upper membrane leaflet revealed a reproducible, heterogeneous distribution of cholesterol around the channel in both simulations.

By using per-residue analyses to quantify occupancy, mean contact duration, simultaneous lipid contacts, pocket stability scores and cholesterol residence times, residue 466 (PHE) was identified as the most consistently contacted residue across all four subunits and both simulation runs. Additional residues, particularly 454 (LEU) and 450 (ILE), showed increased interactions in specific subunits. In contrast, residue 456 (ALA) consistently showed minimal involvement in cholesterol binding throughout the simulations.

Overall, these results support evidence for the existence of a preferred cholesterol interaction region within the transmembrane domain of Kv10.1, centered primarily on residues 466 (PHE), 454 (LEU) and 450 (ILE). Furthermore, these findings support the hypothesis that cholesterol binding at this region may modulate channel gating and could be relevant in the context of altered membrane composition in tumor cells.

Table of contents

Abstract.....	0
Table of contents	2
List of Tables	4
List of Figures	5
1. Introduction.....	6
1.1 Structural basis and physiological role of the Kv10.1 potassium channel	7
1.1.1 Voltage-gated potassium channels: Structural and functional principles	7
1.1.2 Kv10.1 Channel: Molecular characteristics and cellular expression	8
1.1.3 Structural Organization of the Kv10.1 channel	9
1.1.4 Biophysical properties and gating mechanisms.....	11
1.1.5 Regulation and pathophysiological relevance of Kv10.1	12
1.2 Membrane environment and lipid-dependent regulation of the Kv10.1 channel...	15
1.2.1 Structural organization and asymmetry of biological membranes	15
1.2.2 Cholesterol and phospholipids as regulators of ion channels	16
1.2.3 Structural determinants and leaflet-specific lipid modulation of Kv10.1	17
1.3 Kv10.1 as drug target and difference to the Kv11.1 channel.....	19
1.3.1 Kv10.1 as a pharmacological target	19
1.3.2 Structural basis of Kv10.1 inhibition.....	21
1.3.3 Structural and functional comparison of Kv10.1 and Kv11.1	25
2. Materials and Methods	27
2.1 Selection of the protein.....	27
2.2 Structural visualization, alignment and identification of cholesterol-interacting residues.....	27
2.3 Analysis of cholesterol in the coarse-grained wild-type Kv10.1 system.....	28
2.4 Disclaimer on the use of artificial intelligence	29
3. Results.....	30
3.1 Structural overview of Kv10.1 cryo-EM structures.....	30
3.2 Structural alignment and identification of cholesterol-interacting residues	31
3.2.1 Structural comparison of the Kv10.1 wild-type model with crystal structures	31
3.2.2 Identification of cholesterol-interacting residues and other cholesterol pockets	33
.....	33

3.3 Cholesterol distribution and membrane behavior	35
3.3.1 Cholesterol density maps	35
3.3.2 Cholesterol count per leaflet over time	36
3.3.3 Cholesterol coverage around Kv10.1 over time.....	37
3.4 Residue-level cholesterol interaction analysis	38
3.4.1 Cholesterol occupancy per residue	38
3.4.2 Mean contact duration	40
.....	41
3.4.3 Mean simultaneous cholesterol contacts	42
3.4.4 Pocket stability score	44
3.4.5 Cholesterol residence time	46
3.4.6 Minimum distance to nearest Cholesterol.....	48
4. Discussion	51
References	54
Appendix	58

List of Tables

Table 1: Comparison of available Kv10.1 cryo-EM structures	30
Table 2: Cholesterol occupancy (%) per residue across all four subunits (SubA–SubD) and both simulation runs (R1, R2).	39
Table 3: Mean cholesterol contact duration (μs) per residue across all four subunits and both simulation runs.	40
Table 4: Mean number of simultaneous cholesterol contacts per residue across all four subunits and both simulation runs.	43
Table 5: Pocket stability score (0–1) per residue across all four subunits and both simulation runs.	45
Table 6: Cholesterol residence time (μs) per residue across all four subunits and both simulation runs.	47

List of Figures

Figure 1: Structural alignment of 8EOW and 5K7L	31
Figure 2: Structural alignment of 8EOW/5K7L and wild-type model	32
Figure 3: Residues and cholesterol pockets visualised in PyMOL of wild-type simulation run 1	33
Figure 4: Distance analysis between residues 466, 465, 462, 457, 456, 454, 453, 450 and cholesterol binding pocket (5743) visualised in PyMOL of wild-type simulation run 1	34
Figure 5: Cholesterol density map comparison of wild-type run1 and wild-type run 2	35
Figure 6: Cholesterol count per leaflet over time (run1 and run2)	36
Figure 7: Cholesterol coverage around Kv10.1 over time (run1 and run2)	37
Figure 8: Comparison of cholesterol occupancy per residue per subunit of Run 1 and Run 2	39
Figure 9: Comparison of mean cholesterol contact duration per residue per subunit of Run 1 and Run 2	41
Figure 10: Comparison of mean simultaneous cholesterol contacts per residue per subunit of Run 1 and Run 2	43
Figure 11: Comparison of the pocket stability score per residue per subunit of Run 1 and Run 2	45
Figure 12: Comparison of the cholesterol residence time per residue per subunit of Run 1 and Run 2	47
Figure 13: Minimum distance to nearest cholesterol of subunit A of Run 1 and Run 2	49
Figure 14: Minimum distance to nearest cholesterol of subunit B of Run 1 and Run 2	49
Figure 15: Minimum distance to nearest cholesterol of subunit C of Run 1 and Run 2	50
Figure 16: Minimum distance to nearest cholesterol of subunit D of Run 1 and Run 2	50

1. Introduction

Structural data from the Kv10.1 cryo-EM structure (PDB: 5K7L), together with computational docking studies, suggest the possible presence of a potential cholesterol interaction at the interface between the pore domain and the voltage-sensing domain. However, it remains uncertain whether this region represents a stable and specific binding pocket under physiological membrane conditions and whether it is consistently occupied across all four subunits of the tetrameric channel.

This thesis aims to characterize this proposed cholesterol interaction in Kv10.1 using coarse-grained molecular dynamics simulations of the full tetramer embedded in a cholesterol-containing lipid bilayer. In particular, it will be investigated if a preferential cholesterol interaction site can be identified from density maps of the upper membrane leaflet and if this pattern is reproducible across independent simulation runs. Furthermore, it will be examined whether the interactions observed near residues highlighted in the 5K7L structure correspond to a stable and well-defined interaction site. Moreover, this work explores if cholesterol interaction patterns differ between the four subunits of the tetramer and if additional interaction sites can be found near the residues of interest.

1.1 Structural basis and physiological role of the Kv10.1 potassium channel

Voltage-gated potassium (Kv) channels play a central role in the regulation of cellular electrical activity. By setting membrane potential and shaping downstream signaling, they influence neurons and muscle cells (Hille, 2001). By contrast, the ether-à-go-go channel Kv10.1, also known as Eag1 or KCNH1, stands out. In several ways, its structure differs from what is typically seen in other Kv channels, and its gating behavior does not always follow the same rules as established models. More unexpectedly, Kv10.1 has been implicated in discussions of cell proliferation and cancer, an association that remains intriguing and, at times, controversial (Pardo et al., 1999).

To better understand these observations, it is necessary to look more carefully at how Kv10.1 is put together and what functional possibilities that architecture might enable. Its structural features appear to influence not just voltage sensing and gating, but also how the channel responds to its membrane environment and to intracellular modulators (Pardo et al., 1999). In that sense, Kv10.1 seems less like a passive pathway for potassium ions and more like a component whose activity reflects its interactions with its molecular environment. This framing may help explain why the channel has attracted attention as a possible drug target (Whicher & MacKinnon, 2016).

In this chapter, I begin by laying out the basic structural and functional principles shared by voltage-gated potassium channels more generally. From there, I narrow the focus to Kv10.1, outlining its molecular characteristics and what is currently known about its physiological roles. Along the way, I highlight specific structure-function relationships that will matter in later sections, particularly when we turn to lipid-protein interactions, molecular dynamics simulations, and pharmacological approaches.

1.1.1 Voltage-gated potassium channels: Structural and functional principles

Voltage-gated potassium channels make up a large and heterogeneous group of transmembrane proteins that allow potassium ions to cross biological membranes in response to changes in membrane potential (Hille, 2001). Although individual family members serve different physiological roles, they appear to rely on a common architectural framework.

Most Kv channels consist of four homologous subunits that assemble into a tetrameric complex, forming a central pore that allows ions to permeate.

Generally, each subunit consists of six transmembrane helices, called S1 to S6. The first four helices form the voltage-sensing domain, while the fifth and sixth helices combine to form the pore domain. The S4 helix carries positive charges that respond to changes in the membrane potential, transmitting these signals to open the pore through conformational rearrangements along the S4 to S5 linker (Hille, 2001; Whicher & MacKinnon, 2016).

The activity of Kv channels is regulated by both intrinsic and extrinsic factors like the composition of the surrounding membrane, interacting subunits, and post-translational modifications. Kv channels are typically associated with controlling action potentials in neurons and muscle cells (Hille, 2001). However, a closer look shows that there are several members, including Kv10.1, that fulfil roles beyond electrical excitability, for example in regulating cell cycle progression and cell proliferation (Luis et al., 2022).

1.1.2 Kv10.1 Channel: Molecular characteristics and cellular expression

Kv10.1, also referred to as Ether-à-go-go 1 (Eag1), belongs to the KCNH family of voltage-gated potassium channels, which includes Kv11 (hERG) and Kv12 subtypes. It was first cloned in *Drosophila melanogaster* and later expressed in *Xenopus* oocytes (Saganich et al., 2001; Warmke & Ganetzky, 1994).

Structurally, each Kv10.1 subunit consists of six transmembrane segments (S1-S6) similar to other voltage-gated potassium channels, with the S1-S4 region forming the voltage-sensing domain, with S4 carrying positively charged residues that shift in response to membrane depolarization. The S5-S6 region combines to form the central pore, while the P-loop regulates potassium selectivity. Kv10.1 possesses large cytoplasmic N- and C-terminal regions making up around 70% of its molecular mass (Luis et al., 2022).

A closer look at the voltage sensor of Eag1 shows that S4 contains six positively charged residues (K327, R330, R333, R336, R339 and K340). Moreover, residues F261 and D264 from S2 and D299 from S3 form a charge transfer center, which facilitates the movement of these positive charges along S4 responding to changes in membrane voltage. There

are several differences to previous Kv structures. For example, the 13-residue linker between S2 and S3 extends into the cytoplasm. Another difference is the short 5-residue loop of the S4-S5 linker. In contrast to Kv1.2–2.1, where the S4–S5 linker acts as a mechanical lever that couples voltage-sensor movement to the S6 helix, the much shorter S4–S5 linker in Eag1 cannot serve this role. Notably, cutting the S4-S5 linker in Eag1, and thus removing the covalent connection between the voltage sensing domain and the pore still results in a functional voltage-gated channel. These differences in the structural organization indicate that the coupling between voltage sensor movement and pore opening and closing operates through a fundamentally different mechanism in Eag1 compared to Kv1.2-2.1 (Whicher & MacKinnon, 2016).

In healthy tissue, Kv10.1 is found predominantly in the central nervous system where it contributes to shaping action potentials shape and regulating neuronal excitability. It is also expressed at lower levels in myoblasts, as well as in the placenta, testes and adrenal glands. In contrast, Kv10.1 is expressed in over 70% of human tumors including many cancer cell lines from different tissue types like neuroblastoma, breast, melanoma, colon, lung, cervical and ovarian cancers where it has been linked to cell proliferation, angiogenesis, as well as cell migration and invasion. This pattern of overexpression has drawn attention to Eag1 as a potential cancer marker (Ouadid-Ahidouch et al., 2016; Rodríguez-Rasgado et al., 2012).

The difference between its limited expression in healthy tissues and frequent overexpression in tumors has made Kv10.1 an appealing target for therapeutic intervention.

1.1.3 Structural Organization of the Kv10.1 channel

From a structural standpoint, Kv10.1 appears to follow the familiar architectural organization of voltage-gated potassium channels. It assembles as a tetramer, with four identical subunits arranged symmetrically around a central ion-conducting pore (Zagotta et al., 2003).

The difference of the architectural plan of Kv10.1 becomes apparent when compared to the 15-residue helical S4-S5 linker of Kv1.2-2.1 where the transmembrane segments are

domain swapped. In those domain swapped channels, the S4-S5 linker is positioned above the S6 from within its own subunit, forming a ring of linker helices wrapping around the S6 helices. The short 5-residue S4-S5 linker of Eag1 is not able to span the 20 Å distance required for this arrangement. As a consequence, the transmembrane segments of Kv10.1 do not form a domain swapped architecture and stay in a non-domain swapped conformation in all channel states. In this conformation, the S4-S5 linker lies above the S6 from a neighboring subunit and contacts between S1 and S5 form the largest interface between the voltage sensor and the pore (Whicher & MacKinnon, 2016).

The N- and C-termini of Kv10.1 are unusually large. The long N-terminal region contains a Per-Arnt-Sim (PAS) domain, which is located at the periphery of the intracellular region and a PAS-cap domain. In contrast, the C-terminal region of Kv10.1 has a cyclic nucleotide binding homology domain (CNBHD) similar to the cyclic nucleotide binding domains found in HCN and CNG channels, which is linked to the pore through a C-linker region. The PAS domains interact with the CNBHD and are located directly underneath the voltage sensing domain, while the N-terminal region of the PAS domain faces towards the voltage sensor (Han et al., 2017; Whicher & MacKinnon, 2016).

However, the Kv10.1 channel does not bind cyclic nucleotides in a functionally relevant way. It rather has a regulatory role only, as interactions between the PAS domain in the N-terminus and the CNBHD in the C-terminus, appear to form a regulatory module that constrains channel gating, effectively linking distant parts of the protein into a single functional unit (Brelidze et al., 2009; Zagotta et al., 2003; Haitin et al., 2013).

Other structural features of the C-terminus are the ciliary localization signal and the tetramerizing coiled-coil domain (TCC). Furthermore, three calmodulin-binding domains (CaMBD) have been identified in both the C- and N-terminal regions (Han et al., 2017).

This structural complexity may help explain the unusual functional features of Kv10.1 and provides a useful framework for understanding its gating behavior and physiological roles, which are discussed in the following chapter.

1.1.4 Biophysical properties and gating mechanisms

Looking at Kv10.1 from a biophysical standpoint, it becomes clear that it behaves differently from the more familiar voltage-gated potassium channels of the Kv1 or Kv2 families.

The central function of ion channels is ion permeation and this process is regulated by the opening and closing of pores. Channel gating is linked to a sensing mechanism that detects changes in transmembrane voltage but may also be configured to respond to intracellular signals like Ca^{2+} , cyclic nucleotides and potentially other cellular factors. In general, potassium channels are highly selective in allowing passage of ions, and this selectivity determines the direction and physiological consequence of current flow. Inward currents of Na^+ or Ca^{2+} increase the intracellular positive charge, thereby creating further depolarization and activation of voltage-gated channels, whereas outward K^+ currents or inward Cl^- currents reduce positive charge and contribute to the termination of electrical excitation (Yellen, 2002).

Kv10.1 channels open in response to positive membrane potentials. A notable feature of these channels is that their activation kinetics are strongly dependent on the membrane potential, with currents developing more rapidly at higher levels of depolarization. As observed in patch-clamp experiments, Kv10.1 becomes activated during prolonged depolarization (Luis et al., 2022).

This observation suggests that voltage sensing on its own may not be sufficient to fully explain the gating behavior of the channel.

At the molecular level, voltage detection in Kv10.1 relies on positively charged residues within the S4 transmembrane segment. Movement of the voltage sensor is transmitted to the pore by a linker that connects the S4 and S5 segments. This linker forms an α -helix that runs parallel to the membrane and is positioned above S6, acting as a mechanical lever that couples voltage-sensor movement to the opening and closing of the channel. Several studies indicate that the link between S4 movement and pore opening is less strong than in more classical Kv channels (Whicher & MacKinnon, 2016). This means that even after the activation of the voltage sensor, the opening of the pore is delayed. These structural constraints may explain why currents of the Kv10.1 develop slowly despite clear depolarization.

Looking more closely at the S4-S5 linker, it becomes apparent that calmodulin (CaM) is able to bind to intracellular domains and promote closure of the potassium pore independently of the voltage sensor's position. This suggests an alternative gating mechanism for Kv10.1. Every Eag1 subunit contains three CaM binding regions which create two binding sites. Upon closer examination of rEag1, it becomes apparent that a single CaM molecule binds three regions. As a result, four CaM molecules are associated with each rEag1 tetramer. Two of these contact regions are located at the C-terminus of the CNBHD and form two α -helices. These helices act in a claw-like manner, capturing the C-lobe of CaM. The first helix binds with relatively low affinity and interacts near the Ca^{2+} binding sites at the top of the CaM C-lobe. In contrast, the second helix shows higher affinity and interacts with the hydrophobic pocket of the CaM C-lobe. Notably, this disruption of the linker helix that normally connects the C- and N-lobes, allows the hydrophobic core of the CaM N-lobe to bind a helix within the PAS domain (Whicher & MacKinnon, 2016).

By binding both the CNBHD and the PAS domain from neighboring subunits, CaM may function like a molecular clamp, drawing the two domains together and thereby altering the orientation of these regions. Under this clamping mechanism, the CNBHD that directly engages CaM would be drawn toward the neighboring PAS domain. Because the CNBHD is connected to S6 via the C-linker, such displacement is likely to result in a change in the channel structure. Movement of the CNBHD toward the PAS domain causes a rotation of the C-linker and the S6 segment, which may promote a 55° bend in a direction that compacts the intracellular helical bundle forming the gate, leading to the closure of the pore (Whicher & MacKinnon, 2016).

This mechanism provides a structural pathway by which CaM binding could influence channel gating without direct involvement of the voltage sensor.

1.1.5 Regulation and pathophysiological relevance of Kv10.1

Besides the gating modulator calmodulin that was described above, there are other factors that have an influence on the regulation of the Kv10.1 channel. The Ca-binding protein S100B, which binds to the same regions as calmodulin would be an example. Other regulating modulators include a low level of extracellular pH, which shifts activation

toward more depolarized membrane potentials and prolongs activation kinetics, as well as extracellular protons that decrease the modulatory function of Mg^{2+} , thereby delaying voltage-dependent activation of Kv10.1 (Ouadid-Ahidouch et al., 2016).

There are several additional interaction partners and modulators of Kv10.1 that have been identified. These include a wide range of proteins that act as channel partners and influence the trafficking and expression of EAG channels, such as the K^+ channel regulator KCR1, CaM-dependent kinase, the scaffold protein 14-3-3, epsin, rabaptin and cortactin. Furthermore, endogenous small molecules like Mg^{2+} , H^+ , H_2O_2 , arachidonic acid, Ca^{2+} , phosphatidylinositol 4,5-bisphosphate (PIP_2) and hormones such as estrogen and progesterone contribute to the regulation of those channels. Additionally before EAG channels are inserted into plasma membrane, they undergo N-linked glycosylation, which also affects their functional expression (Han et al., 2017).

As mentioned in previous chapters, Kv10.1 is expressed in over 70% of human tumors including many cancer cell lines derived from diverse tissue types like neuroblastoma, breast, melanoma, colon, lung, cervical and ovarian cancers where it has been associated with cell proliferation, angiogenesis, as well as cell migration and invasion. This overexpression has led to the suggestion of Eag1 as a potential cancer marker (Rodríguez-Rasgado et al., 2012).

Studies of cervical biopsy samples first suggested that Eag1 might serve as an early tumor marker. In these studies, Eag1 mRNA was detected in cervical biopsies from patients with normal Pap smear results. Closer examination revealed that one of these patients was infected with human papillomavirus (HPV), the primary etiological factor in cervical cancer. Another patient was diagnosed with an ovarian tumor, while a third showed endometrial hyperplasia. The presence of Eag1 expression has led to the proposal of Eag1 channels as early tumor markers. Furthermore, later in vitro studies have shown that HPV oncogenes might have an influence on the regulation of Eag1 expression (Rodríguez-Rasgado et al., 2012).

In addition to its use as a cancer marker, some studies indicate the use of Eag1 as a prognostic marker across different cancer types. For instance, in colon cancer, patients with higher Eag1 expression exhibited lower survival rates in comparison to patients without Eag1 amplification. A similar association has been reported in patients with acute myeloid leukemia, where Eag1 amplification was related to increasing patient age, elevated relapse rates and significantly reduced survival. Likewise, high levels of Eag1

expression in ovarian cancer patients has been linked with poorer survival, higher tumor grade and the presence of residual disease (Rodríguez-Rasgado et al., 2012).

1.2 Membrane environment and lipid-dependent regulation of the Kv10.1 channel

This chapter provides an overview of how ion channels function within the complex environment of biological membranes. The membrane itself is often seen as a passive component rather than an active regulatory factor. In fact, cellular membranes are chemically diverse and structurally asymmetric, with lipid composition varying between leaflets and across membrane domains. Because of these variations, membrane proteins, including voltage gated potassium channels like the Kv10.1, behave in a different way than usual.

In general, lipids serve three main functions. First, they act as energy stores, providing reserves that are required for processes such as membrane biogenesis. The second function is the formation of the structural framework of cellular membranes by polar lipids. These molecules contain hydrophobic, as well as hydrophilic regions and spontaneously self-associate in aqueous environments. Hydrophobic molecules cluster together to minimize contact with water, while the hydrophilic regions interact with the surrounding aqueous environment and with each other. This amphipathic principle shows the spontaneous formation of lipid bilayers and allows early cells to separate their internal contents from the external environment. Lastly, lipids contribute to cellular signals as first and second messengers in signal transduction and molecular recognition pathways (Van Meer et al., 2008).

Based on these fundamental roles, the next chapter focuses on the structural organization of biological membranes, with emphasis on lipid asymmetry between membrane leaflets and how these features shape membrane function.

1.2.1 Structural organization and asymmetry of biological membranes

As mentioned before, biological membranes are often considered as simple lipid bilayers, but there is more to their structural and functional complexity. Basically, membranes consist of amphipathic lipid molecules arranged into a bilayer, with hydrophobic regions forming an internal core and polar head groups facing the aqueous medium. This arrangement creates a permeability barrier while allowing membrane proteins and ion channels to operate within a flexible, dynamic environment (Hille, 2001).

The main structural lipids in eukaryotic membranes are glycerophospholipids, consisting of phosphatidylcholine, phosphatidylethanolamine, phosphatidylserine, phosphatidylinositol and phosphatidic acid. All of these lipids have a hydrophobic diacylglycerol portion, composed of fatty acyl chains that vary in length and may be either saturated or cis-unsaturated. Interestingly, phosphatidylcholine is found in over 50% of the phospholipids in eukaryotic membranes. It self-associates as a planar bilayer with the hydrophobic tails facing each other and the polar headgroups interacting with the water, creating an almost cylindrical molecular geometry. In contrast, phosphatidylethanolamine forms a conical molecular geometry because of its reduced size of polar headgroups. When phosphatidylethanolamine is included in phosphatidylcholine bilayers, it induces stress onto the curvature of the membrane, which can be utilized for budding, fission and fusion (Van Meer et al., 2008).

Another important aspect of biological membranes is lipid asymmetry. The inner and outer leaflet differ in lipid composition. When examining the glycerophospholipids, it becomes clear that phosphatidylcholine (PC) is located primarily in the outer leaflet of the membrane, whereas aminophospholipids such as phosphatidylserine (PS) and phosphatidylethanolamine (PE) are located on the cytosolic side. Minor components including phosphatidylinositol (PI) and phosphatidic acid (PA) usually are more enriched in the inner leaflet. By contrast, the sphingolipids, sphingomyelin (SM) and glycosphingolipids appear mainly in the outer leaflet. Lipid movement between inner and outer leaflet is known as *flip-flop* (Ikeda et al., 2006).

This asymmetry has functional consequences for signaling, membrane charge and protein localization.

1.2.2 Cholesterol and phospholipids as regulators of ion channels

Upon examining membrane lipids, it becomes clear that they shape the physical environment in which channels operate and might interact with them directly. Two major classes of lipids are cholesterol and phospholipids. While phospholipids influence the baseline properties of the bilayer, cholesterol acts as a modifier, changing membrane mechanics and lipid structure. Both are discussed in the following section.

When looking at phospholipids, they play an important role in forming the structural backbone of the bilayer. Several factors such as head groups, acyl chain length and

degree of saturation contribute to bilayer thickness, curvature stress and lateral pressure profiles. These features can change how transmembrane helices tilt, rotate or shift position during the opening and closure of channels.

Cholesterol contains a planar, rigid steroid ring and a short hydrocarbon tail that is able to intercalate in a lipid bilayer with ease. Cholesterol can change membrane protein structure and function in two ways. It can either bind directly to specific binding sites on the protein or impact different biophysical properties of the bilayer. When adding cholesterol into a lipid bilayer, its fluidity, thickness, lateral pressure profiles and the distribution of surface charges is altered. One example of how a different composition of the membrane can influence protein function can be found in the visual system. An increase in cholesterol in retinal membranes reduces the free volume of the acyl chain packing, thereby shifting the conformational balance between two states in the rhodopsin photocycle (Andersen & Koeppel, 2007; Tillman & Cascio, 2003).

Another aspect of how cholesterol influences ion channels is their association with membrane domains high in cholesterol, often referred to as *lipid rafts*. They are generally described as small, dynamic membrane domains enriched in cholesterol and sphingolipids that help in organizing cellular processes. A number of ion channels have been found within these regions as well as different regulating mechanisms, such as direct binding of cholesterol, an increase in membrane stiffness due to a more tight lipid packing and a concentration of signaling proteins that are able to alter channel behavior. To conclude, lipid rafts add another layer of regulation to how ion channels or other membrane proteins can be influenced by cholesterol (Levitan et al., 2010).

1.2.3 Structural determinants and leaflet-specific lipid modulation of Kv10.1

Kv10.1 is often described mainly as a voltage sensing channel, yet its structural organization suggests that lipid interactions play an important part of its functional regulation. The exposure of the channel to the surrounding bilayer and the many conformational changes happening during gating mechanisms at the protein-membrane boundary show the relevance of interactions between the channel and surrounding lipids. As discussed in the chapters before, the Kv10.1 voltage-sensing domain adopts a non-domain-swapped configuration, which increases the area of direct contact between the voltage-sensing domain and the pore domain, thereby extending the protein-lipid interface (Whicher & MacKinnon, 2016).

Furthermore, the large intracellular domains of Kv10.1 might alter activation kinetics upon movements in these domains. Because of their close position to the inner leaflet, changes in cytoplasmic leaflet composition could possibly affect their interaction networks. This would represent another pathway for leaflet-specific lipid effects (Haitin et al., 2013).

Looking more closely into the KCNH family channels, it becomes evident that Kv10.1 may be sensitive to sterol-rich membrane environments, although experimental evidence remains limited. The transmembrane of Kv10.1 segments show potential cholesterol-recognition patterns, although experimental validation remains limited. Moreover, functional observations reveal that alterations in cholesterol within the membrane affects currents in Eag channels expressed in heterologous systems (Han et al., 2017; Levitan et al., 2010).

1.3 Kv10.1 as drug target and difference to the Kv11.1 channel

The unusual gating profile of Kv10.1 has made its relevance shift from one of many voltage-gated potassium channels to an interesting therapeutic target in the context of cancer biology. This transition happened because of a number of observations that Kv10.1 expression appears to be elevated in a lot of tumor types while remaining relatively restricted in most healthy tissues. This expression pattern makes it an appealing target for drugs with a potential use against various cancer types (Han et al., 2017; Hemmerlein et al., 2006).

One of the challenges regarding the development of specific drugs that target Kv10.1 is its placement within the KCNH family alongside Kv11.1 (hERG), a highly pharmacologically sensitive ion channel. Many molecules that could potentially inhibit Kv10.1 might also influence Kv11.1, making pharmacological separation difficult as it reduces selectivity and safety of such drugs (Vandenberg et al., 2012; Whicher & MacKinnon, 2016).

This chapter examines the pharmacological possibilities of Kv10.1 as a drug target and compares it with Kv11.1 in regard to differences in channel structure and functional behavior.

1.3.1 Kv10.1 as a pharmacological target

Kv10.1 or Eag1 has drawn increasing attention as a potential pharmacological target, mainly because its expression pattern in tumors is different when compared with healthy tissues. There have been several reports that detected Eag1 protein expression in a variety of cell lines derived from human malignant tumors, including neuroblastoma, melanoma, breast cancer and cervical carcinoma. This presence in malignant cells has led to the assumption that Eag1 increases the proliferation of these cells and is essential for maintaining growth (Hemmerlein et al., 2006).

There are many reasons why Eag1 is seen as an appealing target for cancer therapy. First of all is its recurring appearance in tumor cells compared to normal tissues and another reason is the accessibility of the membrane protein from the extracellular side. Unfortunately, it is not enough to see this expression pattern only in cell lines, but the same overexpression needs to be confirmed in actual samples from human tumors. Therefore, immunohistochemical analyses and real-time PCR experiments were

performed to determine Eag1 expression within normal and neoplastic tissues. These studies reported minimal expression in healthy human tissues, while an overexpression in multiple types of human malignant tumors was found (Hemmerlein et al., 2006).

For example, the quantification of the expression of Eag1 within breast carcinomas with the use of real-time PCR showed elevated RNA levels of Eag1 in tumor samples compared to tumor-free tissues. Furthermore, after immunohistochemical analysis of breast carcinomas, it was reported that seven out of eight specimens showed clear Eag1 positivity, although the intensity varied between samples. Moreover, when examining a multi-tissue array comprising samples from fifty breast cancer cases, all samples showed a positive result, with Eag1 detectable in a substantial fraction of the tumor cells. Additionally, results of samples of epidemiologically relevant tumors, such as prostate carcinoma, reveal strong Eag1 positivity in 54 out of 56 cases. A similar outcome was seen in tissues from the colon and the lung. The analysis presents negative or faintly positive results in normal colon epithelium compared to highly positive results in six of eight colon carcinoma samples. Considering lung tissue, some Eag1 expression could be observed in the submucosal glands, while the bronchial epithelium was negative. By comparison, nine out of ten samples of bronchus carcinoma were clearly Eag1 positive. Interestingly, the methods used were unable to detect any Eag1 expression in the liver at the RNA level, although ten out of ten liver carcinomas tested positive. (Hemmerlein et al., 2006).

If you look closely into recent findings about the abnormal upregulation of Kv10.1 in tumor cells it becomes evident that it changes well-known regulatory pathways which include p53, E2F1 and the microRNA miR-34a. The KCNH1 promoter region contains elements that respond to E2F1, while the 3' untranslated region includes binding sites for miR-34a. Functionally, E2F1 appears to increase the expression of Kv10.1, whereas miR-34 acts as a suppressor of the channel. Normally, p53 would keep Kv10.1 levels low by increasing miR-34a expression, which results in suppression of E2F1 and KCNH1 mRNA. However, when the function of p53 is altered, the E2F1 expression is increased while the miR-34a levels decrease, resulting in a positive feedback loop that favors Kv10.1 expression (Pardo & Stühmer, 2014).

This regulatory pattern is consistent with reports that estrogen and the human papilloma virus (HPV) affect Kv10.1 expression, because of the interaction of the estrogen receptor with p53. In contrast, HPV suppresses p53 and stimulates E2F1, which would indirectly

cause higher levels of Kv10.1. A related pathway involves insulin-like growth factor 1 (IGF-1), which has been shown to increase Kv10.1 expression through activation of the kinase AKT1. Notably, AKT1 seems to counteract p53-driven apoptosis, making the suggestion that p53 and Kv10.1 act in opposing directions when cells are pushed toward either apoptosis or proliferation, plausible (Pardo & Stühmer, 2014).

Looking at the microRNA level, additional layers of control have been identified. In glioblastoma, drug-resistant cells show reduced levels of miR-296-3p, which increases Kv10.1 expression. More important, miR-296 mRNAs are also linked to the p53-p21 signaling axis that controls the cell cycle (Pardo & Stühmer, 2014).

Another possible role of Kv10.1 regarding tumor biology is its connection with angiogenesis.

Experimental data suggests that its expression can affect the progression of the G1 phase of the cell cycle and as a consequence directly result in an increase of tumor cell proliferation. One proposed mechanism involves the hypoxia pathway. Kv10.1 expression appears to raise basal expression levels of hypoxia-inducible factor 1 α (HIF-1 α) and makes cells more responsive to mild hypoxia. In practical terms, cells expressing Kv10.1 can activate HIF-1 when there is a minimal drop in the partial pressure of oxygen. HIF-1 then drives the transcription of several downstream factors, which include VEGF, a key promoter of new blood vessel formation. In line with this model, tumor cells that express Kv10.1 have been reported to show an increase in vascularization. These effects would in principle help tumors to become more resistant to low oxygen conditions (Pardo & Stühmer, 2014).

Taken together, Kv10.1 presents an advantage to cancer cells and limiting this would result in an inhibition of tumor progression. The expression patterns make it a promising candidate as a cancer target, however selectivity might complicate development of drugs against this channel.

1.3.2 Structural basis of Kv10.1 inhibition

When Kv10.1 is viewed from a structural perspective, the inhibition of the channel is commonly explained using the same general framework proposed for other voltage-gated potassium channels, in which small molecules enter the inner cavity and stabilize the

channel in a non-conducting state. There are several classes of possible inhibitors, which include classical pore blockers, structure-guided small molecules and antibody-based inhibitors. This chapter will provide a selection of those inhibitors and explain their mechanisms.

Some early structures that were identified as Kv10.1 inhibitors are the tricyclic antidepressant imipramine and the antihistamine astemizole. They were shown to block Kv10.1 currents in heterologous systems in the micromolar range. Electrophysiological data points to a state-dependent block, which correlates with the principle that the inhibitor can reach the inner cavity only after an opening of the gate. The block of the currents is performed by the charged forms of imipramine and astemizole by linking with the intracellular binding sites. This confirms the idea that Eag1 channels seem to have an auxiliary hydrophobic binding site in the intracellular region of the channel. However, these molecules also inhibit Kv11.1, which limits their usefulness as selective drugs. Nevertheless, these early blockers helped establish the drug accessibility of the inner pore of Kv10.1 (García-Ferreiro et al., 2004).

The similarity of the central cavity structures of Kv10.1 and hERG channels has made it difficult to design small molecules with high selectivity for Kv10.1 without affecting hERG. For that reason, attention has increasingly shifted towards alternative strategies. For instance, targeting the voltage-sensing domain instead of the pore itself.

There are a number of small molecules, including mibefradil, purpurealidin I analogues, amiodarone and 20(S)-ginsenoside Rg3 that interact with this domain and modify the gating mechanism. The precise binding sites for these compounds are not known yet, however more recent cryo-EM structures of Kv10.1 and hERG are expected to make structural design easier and may help in developing new specific inhibitors that bind to the voltage-sensing domain (Gubič et al., 2022).

Later on, researchers began using more focused strategies based on structural information and pharmacophore models to develop new inhibitors. One example that was discovered by a ligand-based drug design strategy is the ZVS-08 compound, identified through computational 3D pharmacophore screening. This compound showed activity as a Kv10.1 inhibitor in the low micromolar range. Further chemical modification of this compound led to diarylamine derivatives, referred to as compound 17a and 18b, that

inhibit the channel more strongly, reaching submicromolar or even nanomolar potency against Kv10.1 currents. When examining compounds 17a and 18b for their antiproliferative activity against the MCF-7 breast cancer cell line, it resulted in both compounds moderately reducing the growth of MCF-7 cancer cells. Taken together, both 17a and 18b were shown to inhibit the growth of cancer cells expressing Kv10.1 and hERG, which led to the suggestion of a potential anticancer effect for these new inhibitors. However, achieving good selectivity over the closely related Kv11.1 channel remained a difficulty. (Gubič et al., 2022).

An example of a natural compound drug that targets Kv10.1 is Procyanidin B1. It is a member of the proanthocyanidin family and is commonly found in many plant-based foods, including fruits, vegetables, nuts and seeds. Particularly, grape seeds, show richness in Procyanidin B1 and have been reported to have no potentially genotoxic or toxic effects. Proanthocyanidins are known for a broad range of pharmacological activities, such as antioxidant effects, free radical scavenging and anti-cancer activity. Nevertheless, the precise mechanism by which they exert anti-cancer effects are still not fully understood (Na et al., 2020).

In whole cell patch clamp experiments to study the concentration dependence of Procyanidin B1 in inhibiting Kv10.1, it was shown that the currents of Kv10.1 were significantly inhibited by 1 μ M Procyanidin B1 and almost fully inhibited at 100 μ M. The data reported that increasing concentrations of Procyanidin B1 resulted in a gradual reduction in both outward and inward Kv10.1 currents, accompanied by a progressive loss of the channel's outward rectification. Furthermore, as Procyanidin B1 has no influence on the elevation of intracellular Ca^{2+} , the suggestion that Procyanidin B1 directly inhibits Kv10.1, rather than acting indirectly by using Ca^{2+} as a second messenger, seems plausible. Interestingly, when Procyanidin B1 was tested for its specificity for Kv10.1, it did not affect the currents of the KCNQ1, hERG and Kir2.1 channels, therefore concluding that Procyanidin B1 may be a potent and specific inhibitor of the Kv10.1 channel. Moreover, the docking analysis suggest that residues I550, T552 and Q557 within the C-linker domain of Kv10.1 form a potential binding pocket that appears to be important for the direct inhibition of Procyanidin B1 (Na et al., 2020).

Regarding the exact membrane receptor by which Procyanidin B1 exerts its effects, they have not been clearly identified. Several studies have explored the anti-tumor effects of proanthocyanidins. For example, Meeran and Katiyar reported that proanthocyanidins

suppress mitogenic and survival signaling pathways in vitro and reduce tumor growth in vivo (Meeran, 2008, as cited in Na et al., 2020).

In other studies, it was demonstrated that these compounds can induce apoptosis and inhibit metastasis in highly metastatic breast carcinoma cells. Additionally, they showed that proanthocyanidins reduce UV-induced oxidative stress and inhibit activation of MAPK and NF- κ B signaling pathways in human epidermal keratinocytes, thereby contributing to anti-cancer effects. Likewise, proanthocyanidins are able to promote apoptosis in human epidermoid carcinoma A431 cells through caspase-3 activation by loss of mitochondrial membrane potential. It is also worth noting that proanthocyanidins can trigger apoptosis via pathways involving p53, Bax and caspase-3 (Na et al., 2020).

To conclude, Procyanidin B1 presents a new, potent selective Kv10.1 inhibitor against tumors, acting through multiple molecular mechanisms, although the specific membrane targets responsible for the effects remain to be clarified.

A conceptually different strategy to inhibit the Kv10.1 channel involves monoclonal antibodies directed against extracellular epitopes of the channel. One example of this is mAb56, which recognizes an extracellular region of Kv10.1 and functionally suppresses its activity in tumor cells. Interestingly, treating cancer cells that express Kv10.1 with antibodies has been linked to changes in mitochondrial function. A proposed explanation is that the activity of Kv10.1 indirectly contributes to mitochondrial stability by helping to maintain membrane potential and by regulating downstream calcium signaling pathways. When the channel is blocked, disruptions in cellular ion balance may shift intracellular calcium dynamics, thereby influencing mitochondrial membrane potential and production of reactive oxygen species. Because cancer cells require a high amount of energy and need large amounts of biomolecules such as proteins and DNA to sustain rapid proliferation, migration, angiogenesis and resistance to cell death, targeting mitochondrial stability appears to be a potentially effective strategy in anti-cancer therapy. Overall, mAb56 demonstrates that inhibiting Kv10.1 leads to cellular effects that extend beyond the simple suppression of outward potassium currents. By targeting the channel from the extracellular side, the antibody does not interact with the inner pore shared with Kv11.1, potentially improving subtype selectivity (Hernández-Reséndiz et al., 2020).

In summary, structural inhibition of Kv10.1 appears to be more complex than a simple pore-block model suggests. There is strong evidence for state-dependent block of the

inner cavity by lipophilic small molecules that access the channel after activation. At the same time, there are additional mechanisms, including modulation of the C-linker region, membrane-partitioning inhibitors and extracellular antibody binding. However, what is still missing is a high-resolution structure of Kv10.1 with a bound inhibitor that clearly defines a drug-binding site. Until then, the mechanistic understanding remains incomplete at atomic detail.

1.3.3 Structural and functional comparison of Kv10.1 and Kv11.1

When considering the pharmacology of Kv10.1, it is almost impossible not to also address Kv11.1 (hERG). Both channels belong to the same KCNH family and share substantial structural similarity, which means that many compounds tend to inhibit both channels. In regard to drug development, this makes creating selective drug inhibitors a major challenge.

Despite their structural resemblance, their physiological roles are very different. Kv10.1 is mainly associated with the central nervous system and tumor biology, while Kv11.1 plays a crucial role in cardiac repolarization. Therefore, inhibition of Kv11.1 is strongly linked to QT prolongation and can trigger torsade de pointes. Because of these serious cardiac risks, even small structural differences between the two channels become highly significant when designing drugs (Pardo & Stühmer, 2014).

Kv11.1 was one of the first voltage-gated potassium channels to be linked directly to cancer. Early studies suggested that the resting membrane potential of neuroblastoma cells is closely associated with their ability to proliferate and that is regulated by the expression of an inwardly rectifying channel, later identified as Kv11.1. Later studies showed that Kv11.1 is strongly expressed in many different types of cancer. It has been most extensively studied in leukemia and colon cancer, but increased expression has also been reported in several other tumor types. Upon examining the inhibition of Kv11.1 function or reducing its expression it became clear that Kv11.1 decreases tumor cell proliferation both in vitro and in vivo. Despite this anti-proliferative effect, the channel attracts more attention primarily because of its involvement with drug-induced cardiotoxicity, as mentioned before. As a result, regulatory agencies require that the therapeutic index is tested for every newly invented drug against Kv11.1 to ensure an

adequate safety margin. This requirement makes it challenging to develop therapies that intentionally target Kv11.1, unless ways are found to selectively affect the pathological form of the channel without disturbing its normal cardiac function. It is worth mentioning, that tumor cells seem to prefer expressing a different splice variant of Kv11.1 and may regulate the channel differently from healthy tissue, which could open new opportunities for more selective therapeutic approaches (Pardo & Stühmer, 2014).

Considering the structural organization of both channels, they follow the canonical six transmembrane segment design typical for voltage gated potassium channels. Each subunit consists of a S1-S4 voltage-sensing domain and a S5-S6 pore domain, assembling into a tetramer with a central ion conduction pathway. Moreover, both contain large intracellular PAS domains and cyclic-nucleotide binding homology domains, which are characteristic for KCNH channels. These intracellular domains contribute to gating and coupling mechanisms (Whicher & MacKinnon, 2016).

2. Materials and Methods

2.1 Selection of the protein

The system used in this study was generated in a previous master's thesis (Rüenauver, 2025), where the coarse-grained membrane system and simulation setup were established. The present work focuses on the analysis of cholesterol interactions within this pre-existing simulation system. All system construction and simulation parameters are described in Rüenauver (2025) and were not modified in this study.

The analysis in this work focuses on the coarse-grained wild-type Kv10.1 system embedded in a lipid membrane and is restricted to cholesterol interactions. In total, data from three simulation runs were available for analysis, however only two were included in the present results, as the system in wild-type run 3 underwent tilting.

The molecular dynamics production trajectories were processed using the [gmj trjconv] tool from GROMACS version 2022.3. In the wild-type membrane system, the protein was centered within the simulation box. System preparation, energy minimization, equilibration, and production simulations were carried out using standard GROMACS tools.

The resulting trajectories were visualized using Visual Molecular Dynamics (VMD) version 1.9.4a57 and PyMOL version 2.3 (Humphrey et al., 1996; Schrödinger, LLC, 2015). Statistical analysis and graphical representation of the results were carried out using Python version 3.12.2 with the MDAnalysis library version 2.9.0 and Matplotlib version 3.8.4 (Hunter, 2007; Michaud-Agrawal et al., 2011; Python Software Foundation, 2024).

2.2 Structural visualization, alignment and identification of cholesterol-interacting residues

Structural alignment of the Kv10.1 crystal structure (PDB: 8EOW) and the cryo-EM structure (PDB: 5K7L) was performed using the [super] and [cealign] algorithm implemented in PyMOL (version 2.3, Schrödinger LLC). Since the coarse-grained representations lacked C α coordinates, backbone (BB) beads were renamed to CA prior to alignment to enable structural superimposition.

Figures were rendered at 300 DPI using PyMOL's ray-tracing engine with an opaque white background and labelled afterwards using Microsoft PowerPoint (Microsoft Corporation, 2024).

To find out, which residues interact with cholesterol PyMOL (version 2.3, Schrödinger LLC) was used. The snapshot of wild-type simulation run 1 (wt_r1_center0.gro) obtained from MARTINI coarse-grained molecular dynamics simulation was loaded and visualized in PyMOL. Residues within a 6 Å radius of any cholesterol molecule (resname CHOL) were selected using the PyMOL selection algebra [byres (resid 449-470 within 6 of resname CHOL) and not resname CHOL].

Contact residues were identified by iterating over backbone beads. To identify all cholesterol molecules near the region of interest, a broader selection radius of 10 Å was applied. Each cholesterol molecule was screened for contacts within 6 Å of residues 449-470.

Furthermore, interatomic distances between backbone beads of contact residues and the hydroxyl head bead of cholesterol were calculated using a custom Python script executed within the PyMOL environment, employing NumPy (C. R. Harris et al., 2020).

2.3 Analysis of cholesterol in the coarse-grained wild-type Kv10.1 system

For this leaflet-resolved cholesterol analysis GROMACS version 2022.3 was used to quantify the distribution of cholesterol molecules around the Kv10.1 ion channel (Abraham et al., 2015). Molecular dynamics trajectories of the Kv10.1 channel embedded in a Martini coarse-grained membrane were analyzed for two wild-type replicas. Prior to analysis, trajectories were centered on the protein and corrected for periodic boundary conditions using the GROMACS tool [gmx trjconv] to ensure a stable membrane orientation.

To assign cholesterol molecules to individual membrane leaflets, the centered trajectory was used. Molecules with $z > 0$ were classified assigned to the upper leaflet, whereas molecules with $z < 0$ were assigned to the lower leaflet. Leaflet-specific index groups were generated using the GROMACS selection tool [gmx select] and time-dependent counts of cholesterol molecules per leaflet were obtained by evaluating the size of these groups over the full trajectory.

For the quantification of cholesterol enrichment around the Kv10.1 channel, cholesterol molecules located within 1.1 nm of the protein were identified using distance-based

selections implemented in [gmx select]. The resulting time series were used to determine leaflet-specific cholesterol occupancy around the channel.

To create two-dimensional density maps of cholesterol around Kv10.1 the GROMACS tool [gmx densmap] was used. Density maps were computed in the membrane plane (x-y) with a grid spacing of 0.2 nm. The resulting maps were exported as XPM matrices and converted to graphical formats for visualization.

All analyses were performed independently for each simulation run and the resulting leaflet-specific cholesterol distributions were compared to evaluate the reproducibility and stability of cholesterol interactions with Kv10.1.

Regarding the per-residue cholesterol contact analysis, all analyses were performed using a custom Python script developed for this thesis. The script used Python version 3.12.2 with the MDAnalysis library (version 2.9.0) to load GROMACS trajectory files and calculate, for each residue of interest the cholesterol occupancy, mean contact duration, mean simultaneous cholesterol contacts, pocket stability score, cholesterol residence time and minimum distance to the nearest cholesterol molecule over the simulation trajectory. A contact was defined as any cholesterol molecule with at least one atom within 6 Å of any atom of the target residue. Visualizations were generated using matplotlib (version 3.8.4) and NumPy (version: 1.26.4) was used for numerical operations (C. R. Harris et al., 2020). The full analysis script is available in the appendix.

2.4 Disclaimer on the use of artificial intelligence

For this thesis, AI-assisted applications were used as supportive tools.

ChatGPT (OpenAI, GPT-4, 2025) was used to develop linguistic variants, revise text paragraphs stylistically, sharpen formulations and improve overall readability of passages. Claude (Anthropic, claude-sonnet-4-5, 2025) was used to assist in the development of the Python script for the per-residue cholesterol interaction analysis of the individual channel subunits (*Anthropic*, 2025; *OpenAI*, 2025).

All scientific content, interpretations and conclusions presented in this thesis are the original work of the author. The AI tools were used solely as writing and programming aids and all AI-generated outputs were reviewed by the author and approved by the thesis supervisor.

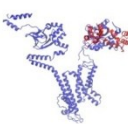
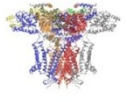
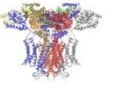
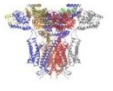
3. Results

3.1 Structural overview of Kv10.1 cryo-EM structures

To place the molecular dynamics simulation results in the context of available experimental data, the four published cryo-EM structures of Kv10.1 were compared in terms of resolution, experimental conditions and sterol content (Table 1).

The earliest structure (PDB: 5K7L) was determined in detergent micelles in the presence of cholesteryl hemisuccinate (CHS) and is the only one showing a partially resolved sterol molecule. In contrast, the three more recent structures (PDB: 8EOW, 8EP0, 8EP1) were obtained in a native-like lipid bilayer containing 5% cholesterol. None of these structures display clearly resolved sterol density (Mandala & MacKinnon, 2022; Whicher & MacKinnon, 2016).

Table 1: Comparison of available Kv10.1 cryo-EM structures

Figure				
PDB ID	5K7L	8EOW	8EP0	8EP1
Resolution (Å)	3.78	3.9	4.9	5.4
Method	Cryo-EM in detergent micelles (DDM/CHS)	Cryo-EM in lipid bilayer vesicles (polarized, ~-145 mV)	Cryo-EM in lipid bilayer vesicles (polarized, ~-145 mV)	Cryo-EM in lipid bilayer vesicles (polarized, ~-145 mV)
Sterol modelled in structure	Yes	No	No	No
CHS used in preparation	Extraction: 0.4% Wash: 0.04% SEC: 0.01%	Extraction: 0.13% Wash: 0.02% SEC: 0.01%	Extraction: 0.13% Wash: 0.02% SEC: 0.01%	Extraction: 0.13% Wash: 0.02% SEC: 0.01%
Cholesterol in final sample	Residual CHS from purification only	5% (wt/wt) cholesterol in POPC:POPG:CHOL (90:5:5) liposomes	5% (wt/wt) cholesterol in POPC:POPG:CHOL (90:5:5) liposomes	5% (wt/wt) cholesterol in POPC:POPG:CHOL (90:5:5) liposomes

3.2 Structural alignment and identification of cholesterol-interacting residues

3.2.1 Structural comparison of the Kv10.1 wild-type model with crystal structures

To validate the structural basis of the molecular dynamics simulations, the wild-type Kv10.1 homology model used for membrane insertion was compared with two experimentally determined structures from the Protein Data Bank. Those were the structures 5K7L, a full-length cryo-EM structure of the *Drosophila* EAG channel and 8EOW, a more recent structure of the human Kv10.1 channel. The structure alignments were performed in PyMOL using the [super] and the [cealign] algorithm.

The alignment between the two reference structures, 5K7L and 8EOW (shown in Figure 1), resulted in an RMSD of 1.913 Å across 5,825 aligned atoms. This relatively low RMSD confirms the strong structural similarity between 5K7L and 8EOW. The overlay shown in Figure 1 shows the alignment of the helical transmembrane segments and intracellular domains.

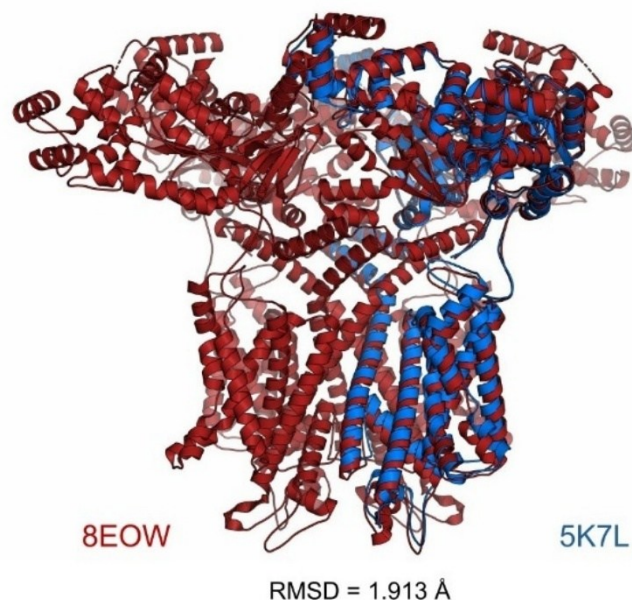


Figure 1: Structural alignment of 8EOW and 5K7L

When the wild-type model was aligned with 8EOW (as shown in Figure 2), a higher RMSD of 3.955 Å was observed over 2,096 residues.

In contrast, the alignment of the wild-type model against 5K7L (Figure 2) returned a lower RMSD of 2.346 Å over 824 residues. This comparison covers the transmembrane and pore-forming regions and suggests that the wild-type simulation model captures the core architecture of the channel.

Overall, these results suggest that the wild-type simulation model is structurally reasonable and consistent with experimentally resolved Kv10.1-family channel structures, especially in the transmembrane region that is most relevant for the potential cholesterol interaction region.

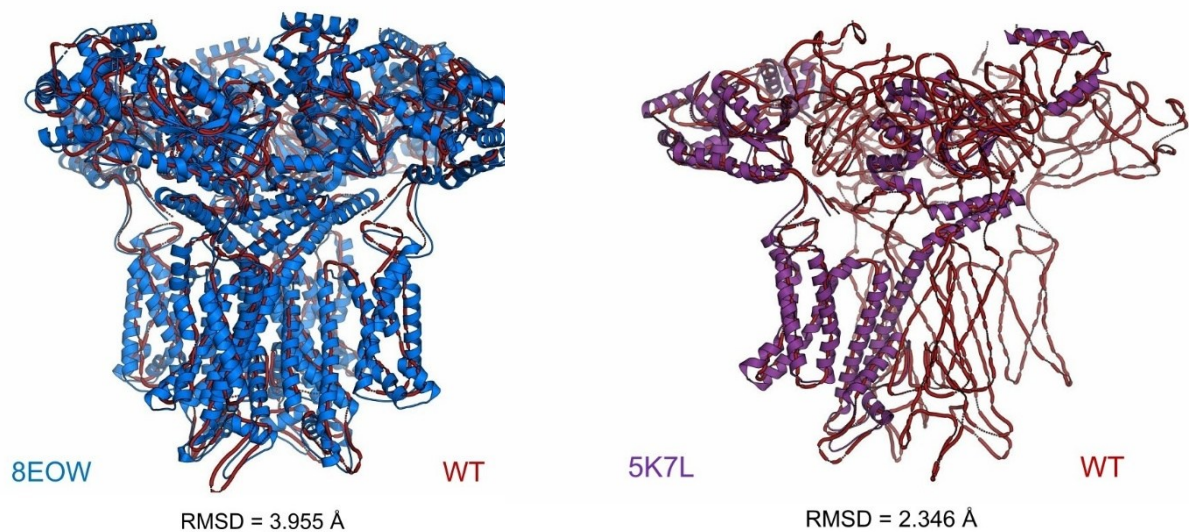


Figure 2: Structural alignment of 8EOW/5K7L and wild-type model

3.2.2 Identification of cholesterol-interacting residues and other cholesterol pockets

To investigate the interaction between cholesterol and the Kv10.1 channel, the end-state structure of wild-type simulation run 1 (wt_r1_center0.gro), was loaded into PyMOL and used as the starting structure for all subsequent cholesterol analyses. A two-step analysis was performed. First, residues within 6 Å of cholesterol molecules were identified. A total of eight residues within the 449-470 segment were found to contact cholesterol, namely residues 450 (ILE), 453 (LEU), 454 (LEU), 456 (ALA), 457 (THR), 462 (VAL), 465 (ILE) and 466 (PHE). Second, a wider search radius of 10 Å was applied to the same region to assess whether additional cholesterol binding sites are present nearby. This revealed two distinct cholesterol pockets. The main pocket, occupied by CHOL 5743, exhibited the highest number of contacts with 19 atoms. An additional pocket that was identified was CHOL 5816 with a number of six contacted atoms. The results of this two-step analysis are presented in Figure 3.

Furthermore, the distance between backbone beads (BB) of each contact residue and the hydroxyl head bead (ROH) of the main cholesterol (residue 5743) were measured (Figure 4).

It should be noted that this analysis was based on a single end-state snapshot and therefore reflects the cholesterol contacts present at one point in time. The eight residues identified here served as candidate positions for the systematic per-residue contact analysis described in chapter 3.4, which uses the full trajectories of two simulation runs.

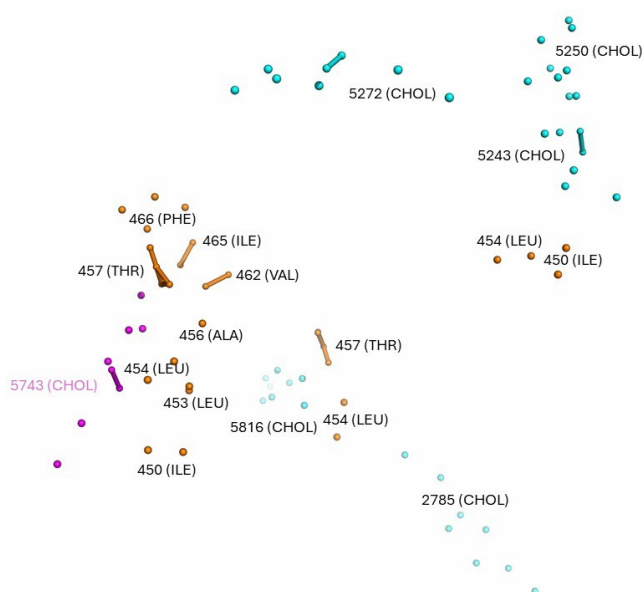


Figure 3: Residues and cholesterol pockets visualized in PyMOL of wild-type simulation run 1

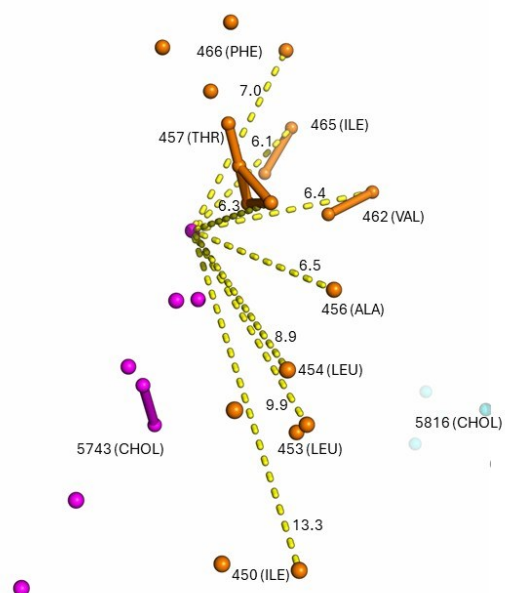


Figure 4: Distance analysis between residues 466, 465, 462, 457, 456, 454, 453, 450 and cholesterol binding pocket (5743) visualized in PyMOL of wild-type simulation run 1

3.3 Cholesterol distribution and membrane behavior

3.3.1 Cholesterol density maps

In contrast to the snapshot-based identification in chapter 3.2.2, the following chapters used the complete trajectories of both simulation runs (run 1 and run 2, each approximately $1.5\mu\text{s}$). To characterize the overall spatial organization of cholesterol in the membrane surrounding the Kv10.1 channel, two-dimensional density maps of cholesterol in the upper leaflet were computed for both replicate runs. The color scale runs from zero density (dark purple, 0.0 nm^{-3}) to high density (yellow, $\sim 0.7\text{ nm}^{-3}$), with the membrane background appearing orange at approximately $0.35\text{-}0.45\text{ nm}^{-3}$.

The density maps (Figure 5) show a consistent reduction in cholesterol directly beneath the channel. In both runs, the central region of the simulation box, corresponding to the transmembrane region of the tetrameric Kv10.1 channel, displays markedly reduced cholesterol density, approaching 0.0 nm^{-3} . This region is surrounded by a broader ring of intermediate density ($0.1\text{-}0.3\text{ nm}^{-3}$). In contrast, areas of the membrane further away from the channel maintain higher density of approximately $0.5\text{-}0.6\text{ nm}^{-3}$.

When run 1 and run 2 are compared, the density maps appear highly similar. Both runs display a central zone of cholesterol depletion surrounded by a ring of intermediate density, as well as a consistent four-lobed pattern at the subunit interfaces. This high level of reproducibility strongly supports the idea that the observed cholesterol distribution reflects a real structural feature of the interaction between cholesterol and Kv10.1.

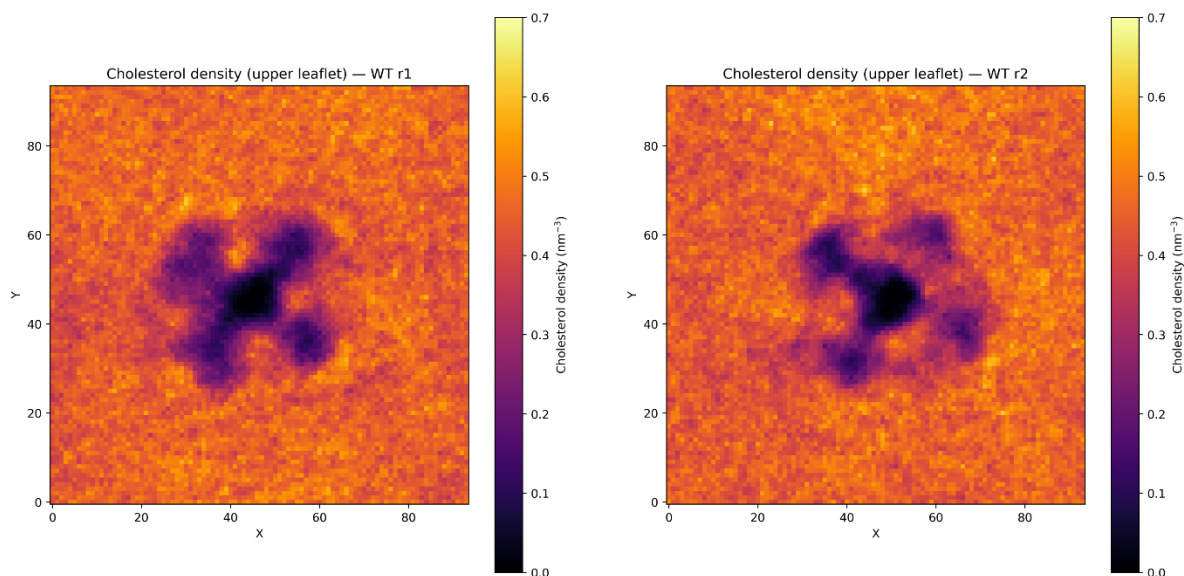


Figure 5: Cholesterol density map comparison of wild-type run1 and wild-type run 2

3.3.2 Cholesterol count per leaflet over time

The number of cholesterol beads assigned to the upper leaflet was monitored across the full trajectory for both simulation runs (Figure 6).

In both runs, the upper leaflet cholesterol count remains at a constant value of approximately 12,464 beads during the entire simulation, displaying a flat horizontal line with no detectable fluctuation.

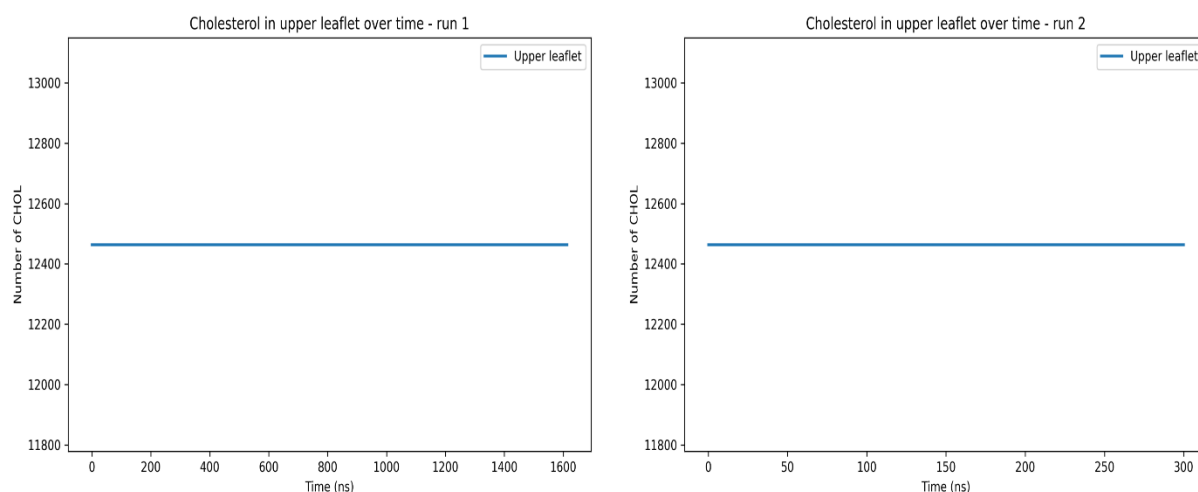


Figure 6: Cholesterol count per leaflet over time (run1 and run2)

3.3.3 Cholesterol coverage around Kv10.1 over time

Figure 7 shows the fraction of total cholesterol beads residing within 6 Å of the Kv10.1 channel surface over the time of each simulation. This metric provides a view of how the protein's cholesterol environment evolves over the course of the simulation.

In both runs, cholesterol coverage begins at approximately 0.012 at the start of the simulation and increases progressively over time.

In run 1, the coverage rises to around 0.012 to plateau at approximately 0.020-0.022 by about 400-500 ns, with continued fluctuations and peaks at ~0.026 at ~1000 ns, after which it partially decreases toward the end of the simulation. In contrast, the increase in run 2 is more monotonic, as coverage rises steadily from 0.012 to about 0.024-0.026 by 1,220-1,600 ns and remains elevated for the rest of the trajectory.

This progressive enrichment in cholesterol near the channel suggests that the membrane is slowly equilibrating over the course of the simulation, with cholesterol slowly diffusing into contact positions with the channel surface. The consistent behavior in both runs indicates that the simulation captures the early stages of cholesterol association with the channel, although longer simulations may be needed to reach full equilibration.

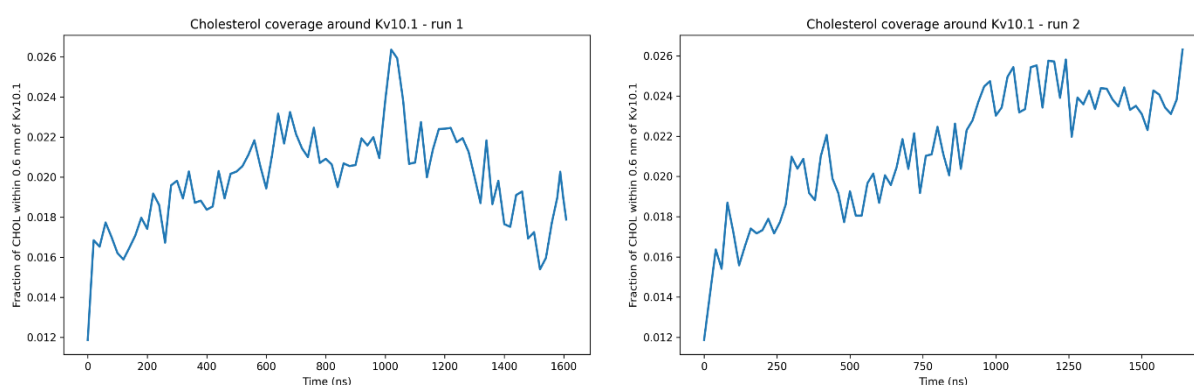


Figure 7: Cholesterol coverage around Kv10.1 over time (run1 and run2)

3.4 Residue-level cholesterol interaction analysis

Per-residue cholesterol contact analysis was performed for eight residues of interest, 450 (ILE), 453 (LEU), 454 (LEU), 456 (ALA), 457 (THR), 462 (VAL), 465 (ILE) and 466 (PHE), across all four subunits (SubA-SubD) and two independent simulation runs of approximately 1.5 μ s each. Contacts were defined using a dual-cutoff scheme (inner cutoff: 4.75 Å, outer cutoff: 6.0 Å). For each residue, occupancy, mean contact duration, mean simultaneous contacts, pocket stability score and cholesterol residence time were calculated and compared between runs.

3.4.1 Cholesterol occupancy per residue

Occupancy expresses the fraction of simulation frames in which at least one cholesterol molecule was detected within 6 Å of a given residue. It indicates how frequently a residue is visited by cholesterol. The occupancy data collected across both runs are presented in Table 2 and Figure 8.

In subunit A, occupancy values are modest across both runs, with residue 457 (THR) showing the highest run 1 occupancy at 32.2% and residue 466 (PHE) at 15.6%. In run 2 these values decrease to 12.5% and 10.4% respectively, suggesting no dominant binding preference at any single region in this subunit.

Looking at the results of subunit B, it shows the highest differences between both runs. In run 1, residue 466 (PHE) displays the highest occupancy with 71.7%, while residue 454 (LEU) remains uncontacted with 0%. In contrast, the pattern in run 2 shows 65.4% occupancy for residue 454 (LEU) and residue 465 (ILE) reaches 52.4%, while occupancy levels for residue 466 (PHE) drops to 16.3%.

When comparing subunit C and subunit D, a similar picture can be seen.

In subunit C, residue 466 (PHE) is unoccupied in run 1 (0%) but rises to 40.9% in run 2, accompanied by a rise in residue 450 (ILE) from 4.2% to 29.3% and residue 454 (LEU) from 2.8% to 33.5%.

In subunit D, residue 466 (PHE) elevates from run 1 with about 14% to ~43% in run 2 and residue 450 (ILE) increases from ~8% to ~23%.

Across all subunits, residue 456 (ALA) consistently shows near-zero occupancy in both runs.

3. Results

Table 2: Cholesterol occupancy (%) per residue across all four subunits (SubA–SubD) and both simulation runs (R1, R2). Values indicate the percentage of trajectory frames in which at least one cholesterol molecule was found within 6 Å of the given residue.

Cholesterol Occupancy (%)								
Residue	Sub A		Sub B		Sub C		Sub D	
	R1	R2	R1	R2	R1	R2	R1	R2
450 (ILE)	6.0	7.5	0.0	14.5	4.2	29.3	7.6	23.2
453 (LEU)	14.0	12.2	0.5	0.7	12.8	0.1	8.1	1.0
454 (LEU)	2.5	2.1	0.0	65.4	2.8	33.5	6.6	14.9
456 (ALA)	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.1
457 (THR)	32.2	12.5	0.0	0.5	11.2	6.9	7.6	8.5
462 (VAL)	0.8	1.1	15.9	0.5	0.0	0.0	5.7	0.7
465 (ILE)	8.1	10.7	13.6	52.4	0.0	0.0	2.1	2.4
466 (PHE)	15.6	10.4	71.7	16.3	0.0	40.9	13.6	43.3

CHOL Occupancy per Residue — Subunit Comparison

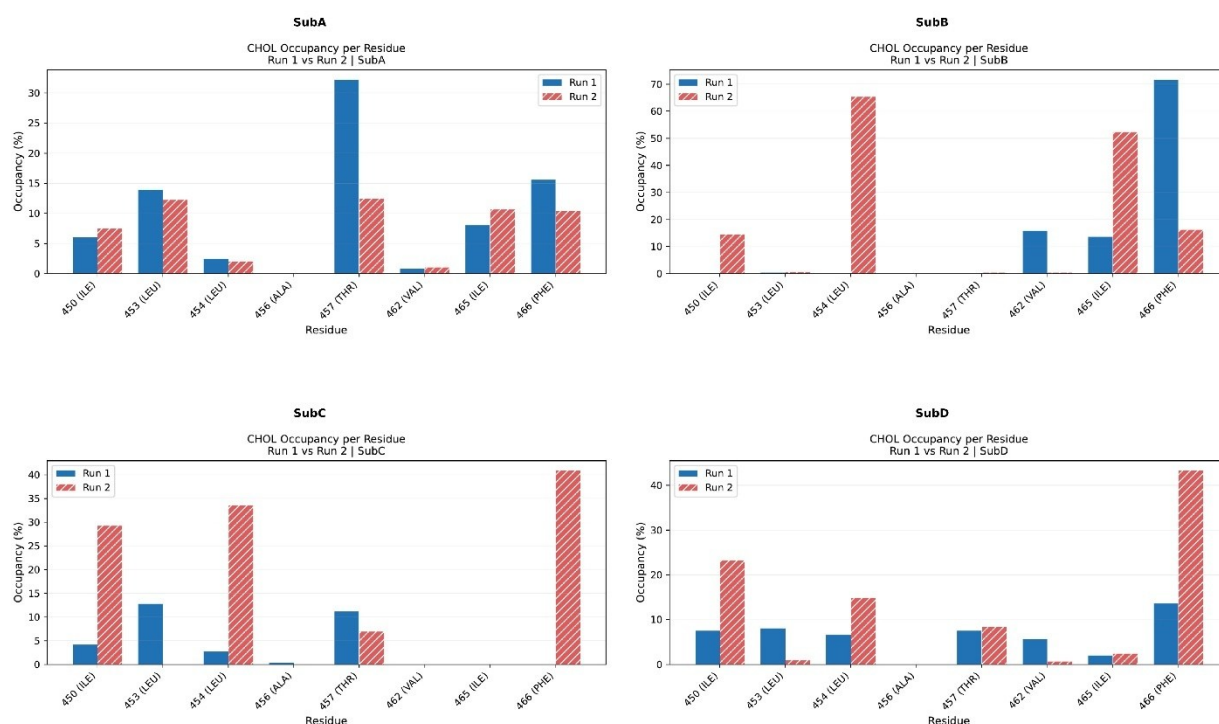


Figure 8: Comparison of cholesterol occupancy per residue per subunit of Run 1 and Run 2

3.4.2 Mean contact duration

The mean duration of individual cholesterol contacts complements the occupancy data by revealing how long individual cholesterol molecules remain in contact with each residue. The results are presented in Table 3 and Figure 9.

In subunit A, residue 457 (THR) and 465 (ILE) show the longest mean contact durations in run 1, at approximately 0.025 and 0.027 μs respectively. In run 2, residue 465 (ILE) increases substantially to 0.044 μs , while the duration at residue 457 (THR) decreases to around 0.011 μs .

Looking at subunit B, residue 454 (LEU) stands out with a mean contact duration approaching 0.077 μs , the longest individual contact duration observed across all subunits and both runs, consistent with its high occupancy of 65.4% under the same condition (Table 2). In run 1, residue 466 (PHE) shows a duration of 0.0371 μs , again mirroring its high occupancy in that run.

In subunits C and D, residue 466 (PHE) in run 2 shows contact durations in the range of 0.019 to 0.022 μs , and residue 450 (ILE) in subunit D run 2 reaches 0.048 μs .

The general pattern suggests that the residues with high occupancy tend to also exhibit longer contact duration, particularly residues 466 (PHE), 454 (LEU), 465 (ILE) and 457 (THR) are capable of sustaining cholesterol interactions over extended timescales.

Table 3: Mean cholesterol contact duration (μs) per residue across all four subunits and both simulation runs. Values reflect the average length of individual uninterrupted contact events (minimum distance < 6 Å).

Mean CHOL Contact Duration (μs)								
Residue	Sub A		Sub B		Sub C		Sub D	
	R1	R2	R1	R2	R1	R2	R1	R2
450 (ILE)	0.0111	0.0138	0.0000	0.0171	0.0117	0.0255	0.0126	0.0480
453 (LEU)	0.0129	0.0144	0.0040	0.0060	0.0141	0.0020	0.0096	0.0023
454 (LEU)	0.0105	0.0085	0.0000	0.0771	0.0092	0.0170	0.0110	0.0226
456 (ALA)	0.0000	0.0000	0.0000	0.0000	0.0030	0.0000	0.0000	0.0020
457 (THR)	0.0254	0.0108	0.0000	0.0020	0.0109	0.0060	0.0053	0.0108
462 (VAL)	0.0035	0.0090	0.0165	0.0040	0.0000	0.0000	0.0078	0.0040
465 (ILE)	0.0268	0.0440	0.0174	0.0787	0.0000	0.0000	0.0057	0.0080
466 (PHE)	0.0117	0.0156	0.0371	0.0208	0.0000	0.0193	0.0151	0.0190

3. Results

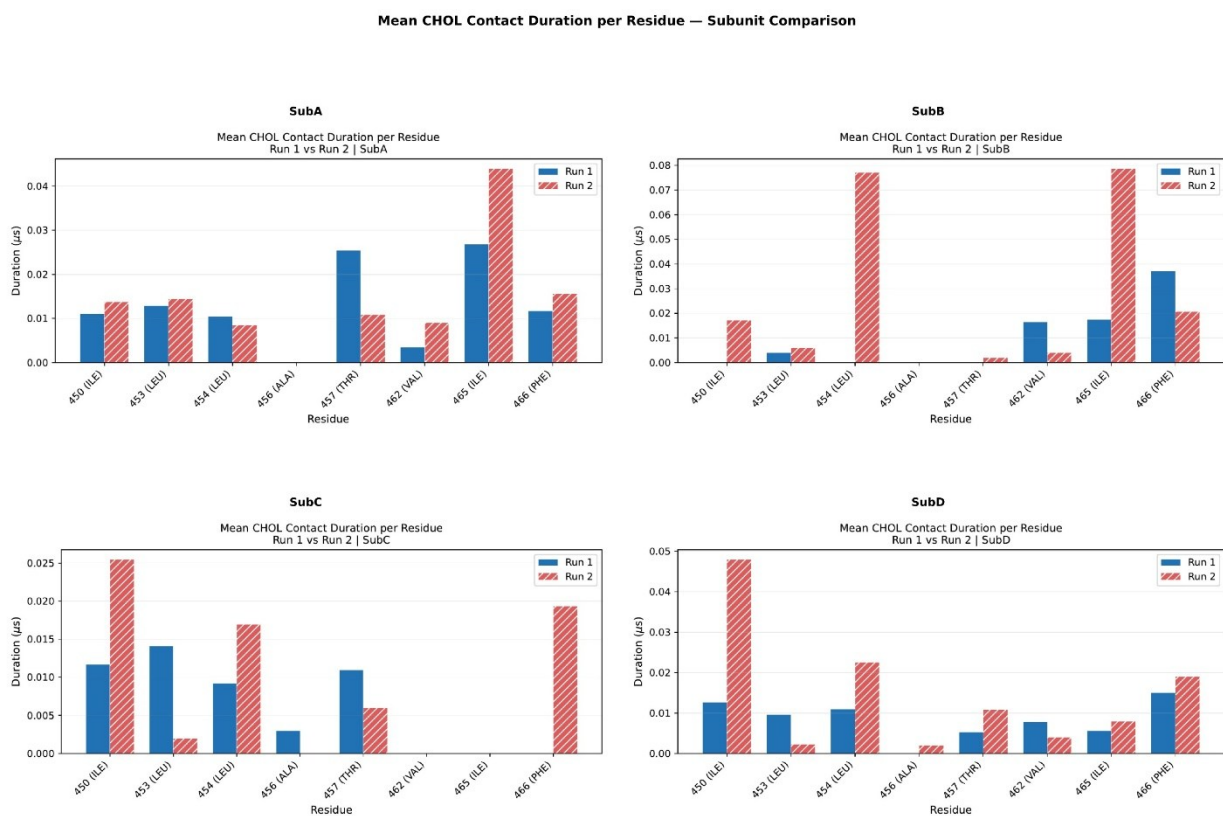


Figure 9: Comparison of mean cholesterol contact duration per residue per subunit of Run 1 and Run 2

3.4.3 Mean simultaneous cholesterol contacts

The mean number of cholesterol beads simultaneously in contact with each residue provides information about the density of cholesterol at the protein surface. Values for all residues and subunits are shown in Table 4 and Figure 10.

For most residues across all subunits, values remain below 0.2 lipids per frame, consistent with the transient nature of lipid-protein interactions in coarse-grained simulations.

However, some residues show noticeably higher contact levels. Residue 466 (PHE) in subunit B run 1 reaches a mean simultaneous contact value of 0.717, the highest recorded across the entire dataset, consistent with its high occupancy in that condition. In subunit B run 2, residue 454 (LEU) reaches 0.654 and residue 465 (ILE) reaches 0.524. In subunits C and D, residue 466 (PHE) in run 2 shows values of around 0.41 and 0.45 respectively. These elevated counts indicate that more than one cholesterol molecule can simultaneously occupy the area of these residues.

Overall, these findings suggest that although most residues interact with only one cholesterol molecule at a time, certain positions, especially residues 466 (PHE) and in subunit B, residues 454 (LEU) and 456 (ALA) can accommodate a higher local concentration of cholesterol. This may indicate the presence of more pocket-like regions at these sites.

3. Results

Table 4: Mean number of simultaneous cholesterol contacts per residue across all four subunits and both simulation runs. Values represent the average number of cholesterol molecules found within 6 Å of the given residue across all trajectory frames.

Mean Simultaneous CHOL Contacts								
Residue	Sub A		Sub B		Sub C		Sub D	
	R1	R2	R1	R2	R1	R2	R1	R2
450 (ILE)	0.060	0.075	0.000	0.145	0.042	0.293	0.076	0.232
453 (LEU)	0.140	0.122	0.005	0.007	0.128	0.001	0.081	0.010
454 (LEU)	0.025	0.021	0.000	0.654	0.028	0.339	0.066	0.150
456 (ALA)	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.001
457 (THR)	0.323	0.125	0.000	0.005	0.112	0.080	0.076	0.085
462 (VAL)	0.009	0.011	0.159	0.005	0.000	0.000	0.057	0.007
465 (ILE)	0.081	0.106	0.137	0.524	0.000	0.000	0.021	0.024
466 (PHE)	0.156	0.104	0.717	0.163	0.000	0.409	0.137	0.449

Mean Simultaneous CHOL Contacts per Residue — Subunit Comparison

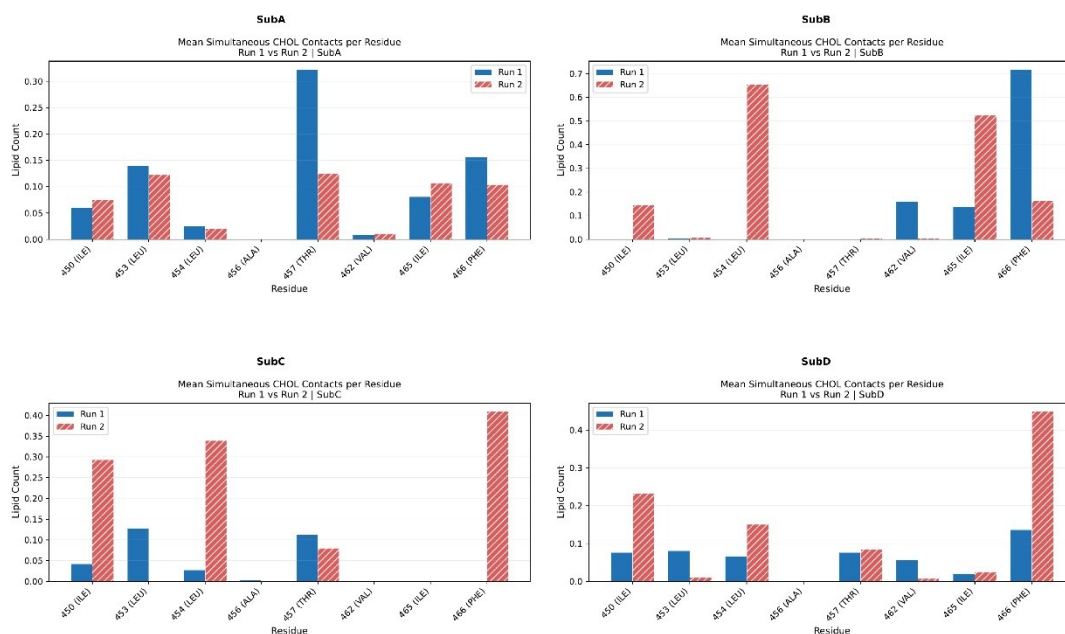


Figure 10: Comparison of mean simultaneous cholesterol contacts per residue per subunit of Run 1 and Run 2

3.4.4 Pocket stability score

The pocket stability score quantifies how consistently a cholesterol-bound state is maintained over the course of the simulation, using a scale from 0 to 1, where 1.0 represents continuous occupancy over the entire trajectory. Results are shown in Table 5 and Figure 11.

Scores below 0.1 are observed for the majority of residues across all subunits, reflecting the transient nature of individual contacts. However, several positions show notably higher stability.

In subunit A, no residue exceeds a stability score of 0.10 in either run, indicating that cholesterol contacts at this subunit are largely transient.

Subunit B shows the highest variation between runs. The highest stability score across the entire dataset is observed at residue 466 (PHE) in run 1 at 0.622, indicating that this position is occupied by cholesterol for more than 60% of the simulation. In run 2, a complementary pattern is seen. Residue 450 (ILE) reaches 0.460 and residue 454 (LEU) reaches 0.413, consistent with the shift in occupancy observed between runs in this subunit.

In subunits C and D, residue 466 (PHE) shows elevated stability in run 2 (subunit C: 0.184; subunit D: 0.407), along with residue 454 (LEU) in subunit C (0.429) and residue 457 (THR) in subunit C run 2 (0.217) and subunit D run 1 (0.267). Residue 456 (ALA) scores zero in all subunits and both runs, confirming its minimal role in cholesterol interaction.

3. Results

Table 5: Pocket stability score (0–1) per residue across all four subunits and both simulation runs. Higher values reflect greater consistency and persistence of cholesterol binding throughout the simulation trajectory.

Pocket Stability Score (0–1)								
Residue	Sub A		Sub B		Sub C		Sub D	
	R1	R2	R1	R2	R1	R2	R1	R2
450 (ILE)	0.000	0.000	0.000	0.460	0.000	0.264	0.000	0.000
453 (LEU)	0.102	0.000	0.000	0.000	0.045	0.000	0.117	0.000
454 (LEU)	0.000	0.000	0.000	0.413	0.000	0.429	0.000	0.000
456 (ALA)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
457 (THR)	0.000	0.027	0.000	0.000	0.003	0.217	0.267	0.000
462 (VAL)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
465 (ILE)	0.000	0.000	0.000	0.176	0.000	0.000	0.000	0.000
466 (PHE)	0.000	0.000	0.622	0.000	0.000	0.184	0.110	0.407

Pocket Stability Score per Residue — Subunit Comparison

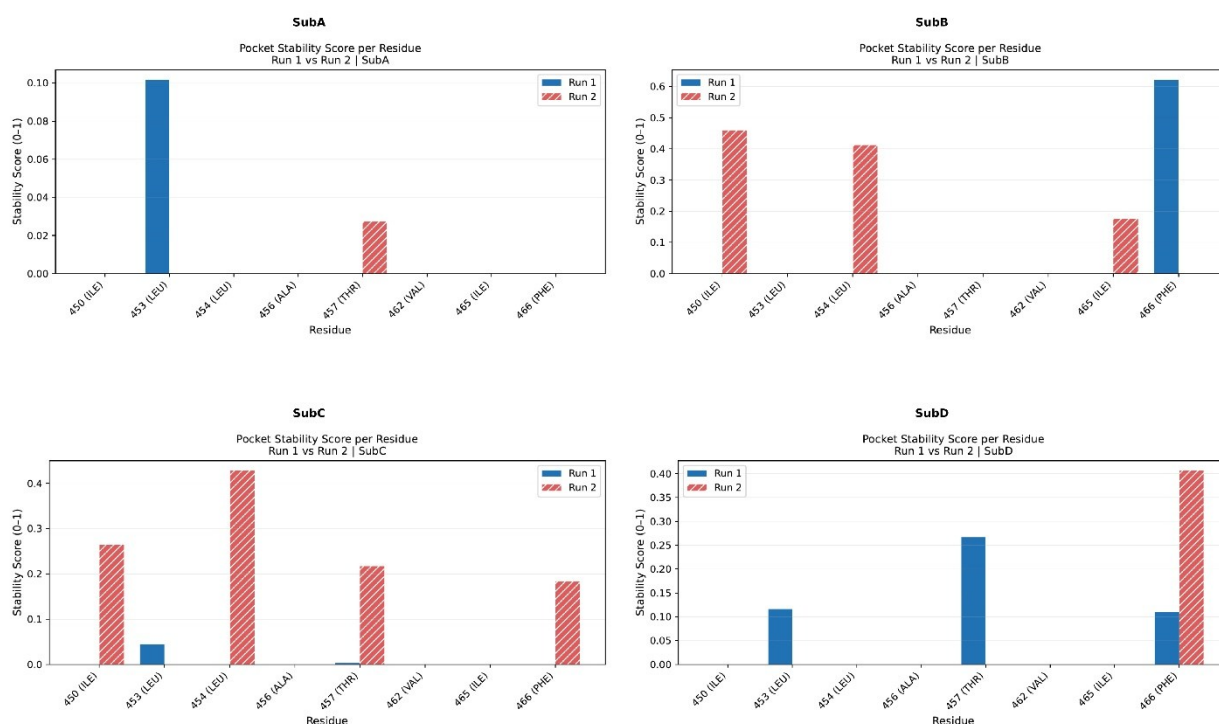


Figure 11: Comparison of the pocket stability score per residue per subunit of Run 1 and Run 2

3.4.5 Cholesterol residence time

Cholesterol residence time reflects the total duration that an individual cholesterol molecule remains continuously associated with a given residue. Results are presented in Table 6 and Figure 12. Most residues show residence times in the sub-microsecond range. A subset of residues, however, record values on the order of 10^{12} μ s, indicating that no dissociation event occurred during the entire simulation. Residues for which no cholesterol contact was recorded in a given run are indicated by a dash in Table 6, as residence time is undefined in the absence of any binding event. This applies to residue 456 (ALA) across all subunits and both runs. If a contact was detected but no dissociation occurred within the simulation timescale, the values are reported as N/A* and should be interpreted as lower bounds of the actual residence time rather exact measurements.

Residues for which no dissociation was observed include 453 (LEU) and 466 (PHE) in subunit A run 1, residue 450 (ILE) in subunit B run 2, residue 457 (THR) in subunit C run 2, residue 462 (VAL) in subunit D run 1, and residue 466 (PHE) in subunit D run 2. For residues where dissociation events were observed, residence time range from around 0.006 μ s to 2.7 μ s, with the longest value recorded at residue 466 (PHE) in subunit D run 1 (2.708 μ s) and residue 457 (THR) in subunit D across both runs (1.134-1.506 μ s).

Taken together, the residence time data indicate that the cholesterol contacts at residues 466 (PHE), 457 (THR) and 450 (ILE) are among the most persistent observed in this dataset.

However, it should be noted that residence time can only be determined reliably when dissociation events are observed within the simulation timescale. Since no dissociation was recorded, the values reported here represent lower bounds rather than true residence times. Consequently, these results cannot be taken as proof of highly specific binding, but indicate that certain cholesterol contacts are stable during the simulation timescale.

3. Results

Table 6: Cholesterol residence time (μ s) per residue across all four subunits and both simulation runs. N/A indicates that no cholesterol desorption event was observed; the value represents a lower bound on the true residence time. — indicates zero occupancy in that run*

CHOL Residence Time (μ s)								
Residue	Sub A		Sub B		Sub C		Sub D	
	R1	R2	R1	R2	R1	R2	R1	R2
450 (ILE)	0.274	0.164	—	N/A*	0.127	0.488	0.321	0.173
453 (LEU)	N/A*	0.356	—	0.060	0.437	—	1.044	0.063
454 (LEU)	0.012	0.020	—	0.739	0.021	0.738	0.187	0.260
456 (ALA)	—	—	—	—	—	—	—	—
457 (THR)	0.254	0.115	—	—	1.134	N/A*	1.506	1.483
462 (VAL)	0.006	0.007	0.195	—	—	—	N/A*	0.002
465 (ILE)	0.067	0.083	0.192	0.543	—	—	0.117	0.020
466 (PHE)	N/A*	0.195	1.010	0.179	—	0.550	2.708	N/A*

CHOL Residence Time per Residue — Subunit Comparison

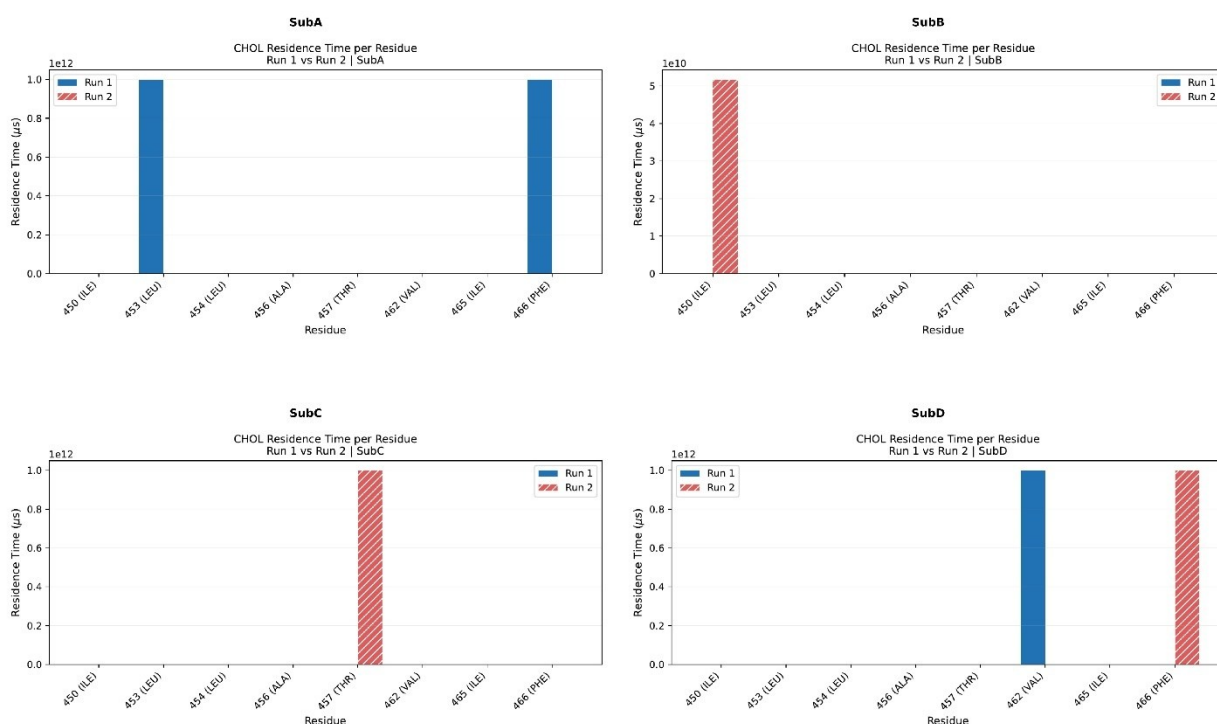


Figure 12: Comparison of the cholesterol residence time per residue per subunit of Run 1 and Run 2

3.4.6 Minimum distance to nearest Cholesterol

The time-dependent analysis of the minimum distances between each residue and the nearest cholesterol molecule, as shown in Figures 13-16, provides a detailed view of how cholesterol approaches and interacts with the channel over the course of the simulation. In all four subunits and both runs, most residues fluctuate between approximately 4 and 16 Å from the nearest cholesterol, with repeated drops below the 6 Å contact cutoff, indicating direct contact events with the occupancy and duration data.

A small number of residues display distance profiles indicative of more stable interactions. Residue 457 (THR) in subunit A run 1 remains within the 6 Å range for extended and largely uninterrupted periods, consistent with its high occupancy of 32.2% in that condition (Table 2). Residue 466 (PHE) in subunit B run 1 similarly maintains prolonged near-contact distances, consistent with its high occupancy of 71.7% (Table 2). Residue 456 (ALA) stands out in the opposite direction, rarely approaching within 6 Å across any subunit or either run, confirming its unimportant role in cholesterol interaction.

However, the most important feature of the minimum distance profiles is the extent of differences between run 1 and run 2. These differences are not limited to individual residues but are apparent across multiple subunits and positions. For example, residue 454 (LEU) in subunit B shows close-contact periods in run 2 that are entirely absent in run 1, and residue 450 (ILE) in subunit C displays a markedly different distance profile between the two runs. These differing trajectories indicate that the distance analyses do not converge toward a consistent model of cholesterol interaction. The profiles that appear stable in one run are generally not reproduced in the replicate run, further supporting the conclusions drawn from the occupancy and stability score data.

3. Results

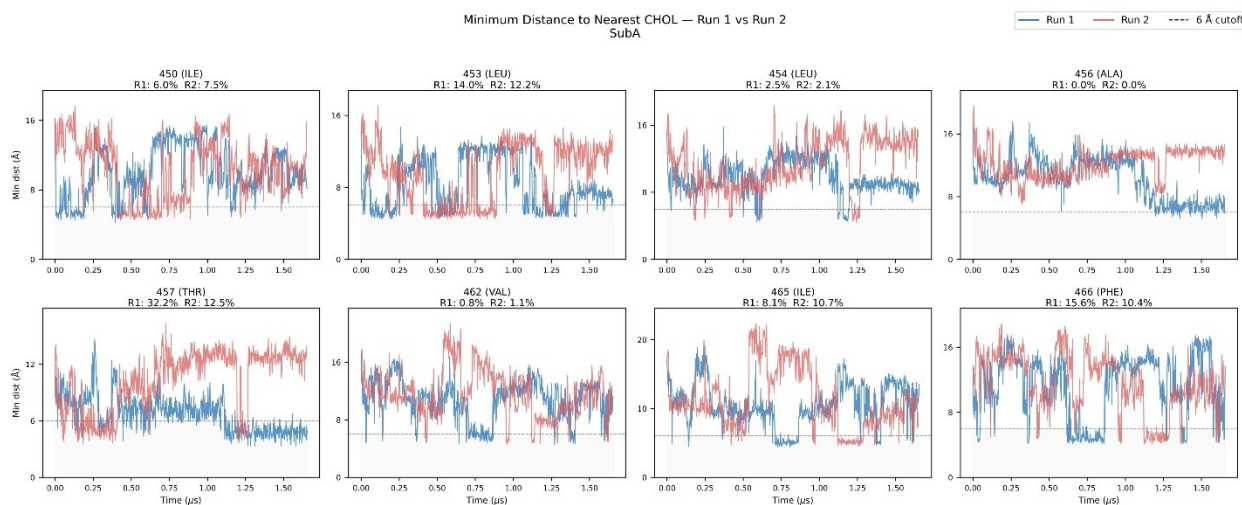


Figure 13: Minimum distance to nearest cholesterol of subunit A of Run 1 and Run 2

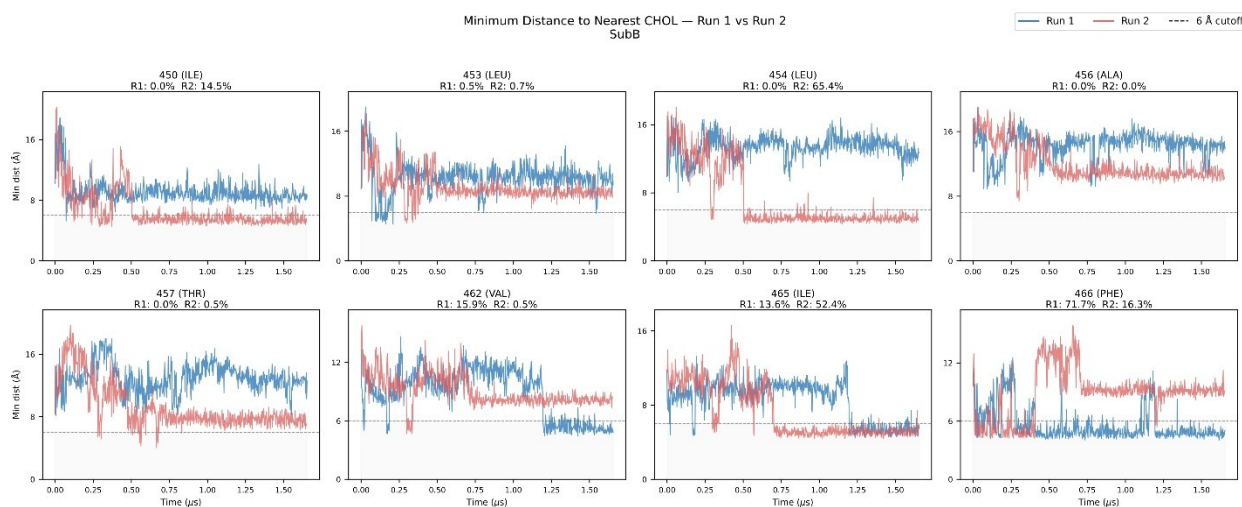


Figure 14: Minimum distance to nearest cholesterol of subunit B of Run 1 and Run 2

3. Results

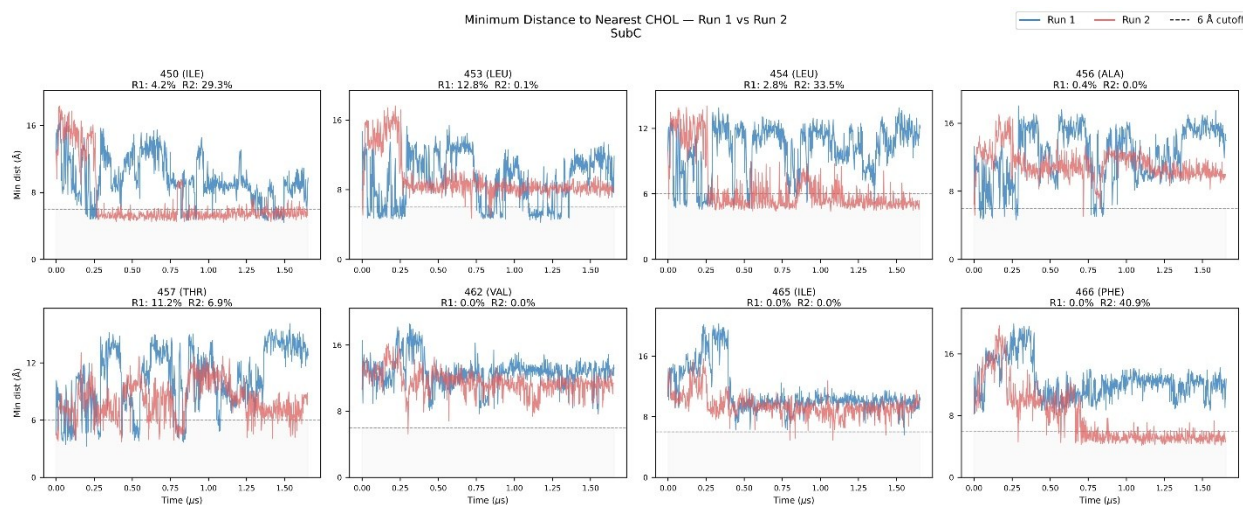


Figure 15: Minimum distance to nearest cholesterol of subunit C of Run 1 and Run 2

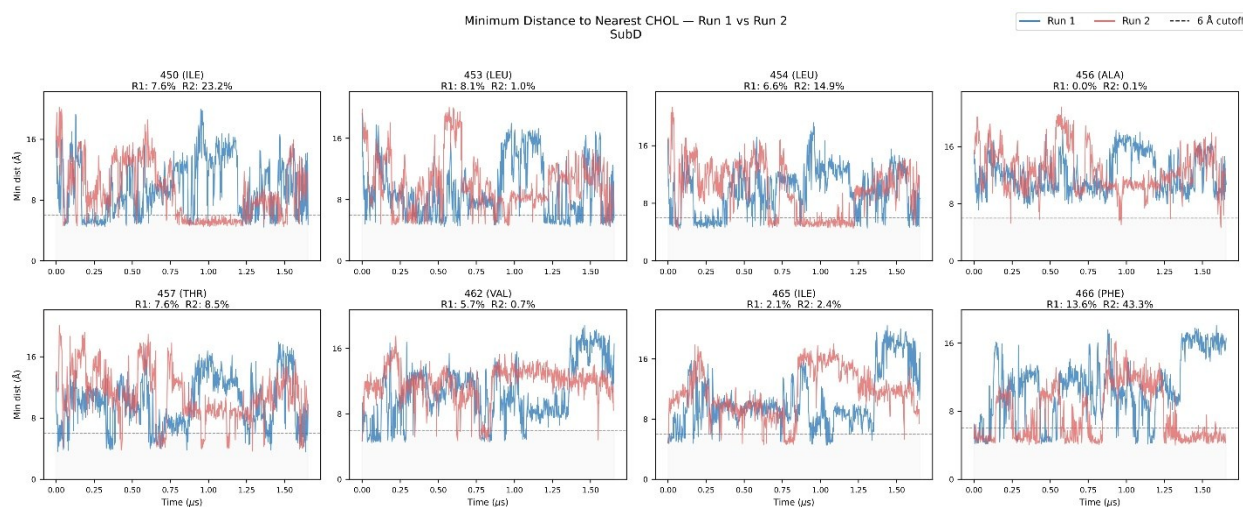


Figure 16: Minimum distance to nearest cholesterol of subunit D of Run 1 and Run 2

4. Discussion

The results obtained from both simulation runs do not provide consistent evidence for a well-defined, specific cholesterol interaction region within the transmembrane region of Kv10.1. While individual interaction events are observed at several residues, the differences across subunits and between runs indicates no existence of such an interaction region.

In both runs, the cholesterol density maps reveal a heterogenous distribution around the protein with zones of locally elevated density surrounding the channel tetramer. While this reproducibility indicates that the overall cholesterol distribution around the channel is not random, density maps reflect bulk lipid organization around the protein rather than specific binding on the residue level. They therefore cannot by themselves confirm the existence of a cholesterol interaction region.

The cholesterol count in the upper leaflet remains constant over the timescale of both simulations and displays a flat horizontal line with no detectable fluctuation. This confirms that there is no systematic change in leaflet composition and indicates that cholesterol flip-flop between the two leaflets does not happen, thereby demonstrating that the membrane bilayer maintains its asymmetric composition throughout both trajectories. This validates the integrity of the simulation setup but does not provide information about specific binding.

The progressive increase in cholesterol coverage around the channel observed in both runs, likely represents an initial phase during which cholesterol molecules diffuse toward the protein surface and become partially equilibrated into preferred positions.

At the per-residue level, the data reveal a highly inconsistent picture across the four subunits and both simulation runs, which is the central finding of this work.

No single residue consistently exhibits consistently high occupancy, long contact duration and an elevated stability score across all four subunits and both runs simultaneously. Residue 466 (PHE) shows the highest individual values in the dataset with an occupancy of 71.7% and a stability score of 0.622 in subunit B run 1, but these are not reproduced in run 2 of the same subunit, where occupancy drops to 16.3% and the stability score falls to zero (Table 2; Table 5). A contact observed in one run but highly absent in a replicate

under identical conditions is more likely due to limited sampling rather than representing a stable, well-defined binding region.

Regarding residue 454 (LEU), a complementary pattern becomes clear, being nearly uncontacted in most conditions but displaying high increases in subunit B and C in run 2, suggesting that cholesterol access to this residue may depend on a specific channel orientation that is more frequently adapted in run 2. Similarly, residue 450 (ILE), shows a general increase in contacts across several subunits in run 2.

Residues 466 (PHE), 454 (LEU) and 450 (ILE) do show the highest contact metrics across the dataset and are located in the same region of the transmembrane domain. However, given the absence of consistent inter-run reproducibility, they cannot be described as defining a cholesterol binding pocket on the basis of the present data alone. The stability scores and residence time data support this. While high stability values and non-dissociating contacts are observed at certain positions, these vary between residues and subunits across both runs. The absence of dissociation at several positions means that true residence times cannot be determined and the reported values should be considered as lower bounds rather than clear evidence of specific binding.

Residues 456 (ALA) and 462 (VAL) consistently show minimal interaction across all subunits and both runs, suggesting that their positions are less favorable for cholesterol interaction. When examining residue 457 (THR), it showed high occupancy in subunit A, run 1 (32.2%), but was much lower in other subunits and run 2 (Table 2). This variability further displays the overall pattern where individual high-contact events occur at different residues under different conditions without a consistent pattern emerging across the dataset.

It should be noted that the electron density observed in the region of interest in the cryo-EM structure (PDB: 5K7L) was not assigned to cholesterol in the original publication and remains of unknown identity. The simulations were designed to test whether cholesterol preferentially occupies this region, rather than to confirm an established interaction site.

Taken together, with the present data no specific, structurally defined cholesterol binding pocket can be confirmed within the 449-470 segment of Kv10.1 on the basis of two simulation runs. The variability across subunits and between runs is too large to indicate a reproducible interaction pattern and likely reflects a combination of asymmetric cholesterol interaction, limited sampling within the simulation timescale and possible differences in the cholesterol distribution around the protein in the two runs.

To conclude, the data of two simulation runs are insufficient to establish statistical significance for lipid contact measurements of this type. Therefore, future work should include additional independent simulation runs to allow better averaging of per-residue contact metrics and to improve inter-run reproducibility. Furthermore, longer trajectories would improve the sampling of cholesterol exchange events. Together, these extensions will be necessary to definitively establish whether the residues identified in this work constitute a true, functionally relevant cholesterol binding pocket for Kv10.1 and to determine how this interaction may contribute to the channel's regulation in the context of cancer biology.

References

- Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., & Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1–2, 19–25. <https://doi.org/10.1016/j.softx.2015.06.001>
- Andersen, O. S., & Koeppe, R. E. (2007). Bilayer Thickness and Membrane Protein Function: An Energetic Perspective. *Annual Review of Biophysics and Biomolecular Structure*, 36(1), 107–130. <https://doi.org/10.1146/annurev.biophys.36.040306.132643>
- Anthropic (2025): Development of Python scripts for per-residue cholesterol interaction analysis. Generative AI chat. Claude (claude-sonnet-4-5). <https://claude.ai>
- García-Ferreiro, R. E., Kerschensteiner, D., Major, F., Monje, F., Stühmer, W., & Pardo, L. A. (2004). Mechanism of Block of hEag1 K⁺ Channels by Imipramine and Astemizole. *The Journal of General Physiology*, 124(4), 301–317. <https://doi.org/10.1085/jgp.200409041>
- Gubič, Š., Toplak, Ž., Shi, X., Dernovšek, J., Hendrickx, L. A., Pinheiro-Junior, E. L., Peigneur, S., Tytgat, J., Pardo, L. A., Peterlin Mašič, L., & Tomašič, T. (2022). New Diarylamine KV10.1 Inhibitors and Their Anticancer Potential. *Pharmaceutics*, 14(9), 1963. <https://doi.org/10.3390/pharmaceutics14091963>
- Haitin, Y., Carlson, A. E., & Zagotta, W. N. (2013). The structural mechanism of KCNH-channel regulation by the eag domain. *Nature*, 501(7467), 444–448. <https://doi.org/10.1038/nature12487>
- Han, B., Tokay, T., Zhang, G., Sun, P., & Hou, S. (2017). Eag1 K⁺ Channel: Endogenous Regulation and Functions in Nervous System. *Oxidative Medicine and Cellular Longevity*, 2017(1), 7371010. <https://doi.org/10.1155/2017/7371010>
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hemmerlein, B., Weseloh, R. M., de Queiroz, F. M., Knotgen, H., Sanchez, A., Rubio, M. E., & Pardo, L. A. (2006). Overexpression of Eag1 potassium channels in clinical tumours. *Molecular Cancer*, 5, 41. <https://doi.org/10.1186/1476-4598-5-41>

- Hernández-Reséndiz, I., Pacheu-Grau, D., Sánchez, A., & Pardo, L. A. (2020). Inhibition of Kv10.1 Channels Sensitizes Mitochondria of Cancer Cells to Antimetabolic Agents. *Cancers*, 12(4), 920. <https://doi.org/10.3390/cancers12040920>
- Hille, B. (2001). *Ion Channels of Excitable Membranes* (3rd edn). Sinauer Associates.
- Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD: Visual molecular dynamics. *Journal of Molecular Graphics*, 14(1), 33–38. [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Ikeda, M., Kihara, A., & Igarashi, Y. (2006). Lipid Asymmetry of the Eukaryotic Plasma Membrane: Functions and Related Enzymes. *Biological and Pharmaceutical Bulletin*, 29(8), 1542–1546. <https://doi.org/10.1248/bpb.29.1542>
- Levitan, I., Fang, Y., Rosenhouse-Dantsker, A., & Romanenko, V. (2010). Cholesterol and Ion Channels. In J. R. Harris (Ed.), *Cholesterol Binding and Cholesterol Transport Proteins*: (Vol. 51, pp. 509–549). Springer Netherlands. https://doi.org/10.1007/978-90-481-8622-8_19
- Levitan, I., Singh, D. K., & Rosenhouse-Dantsker, A. (2014). Cholesterol binding to ion channels. *Frontiers in Physiology*, 5, 65. <https://doi.org/10.3389/fphys.2014.00065>
- Luis, E., Anaya-Hernández, A., León-Sánchez, P., & Durán-Pastén, M. L. (2022). The Kv10.1 Channel: A Promising Target in Cancer. *International Journal of Molecular Sciences*, 23(15), 8458. <https://doi.org/10.3390/ijms23158458>
- Mandala, V. S., & MacKinnon, R. (2022). Voltage-sensor movements in the Eag Kv channel under an applied electric field. *Proceedings of the National Academy of Sciences*, 119(46), e2214151119. <https://doi.org/10.1073/pnas.2214151119>
- Meeran, S. M. (2008). Proanthocyanidins inhibit mitogenic and survival-signaling in vitro and tumor growth in vivo. *Frontiers in Bioscience*, 13(13), 887. <https://doi.org/10.2741/2729>
- Michaud-Agrawal, N., Denning, E. J., Woolf, T. B., & Beckstein, O. (2011). MDAAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, 32(10), 2319–2327. <https://doi.org/10.1002/jcc.21787>
- Microsoft Corporation. (2024). *Microsoft PowerPoint* (Version 2408 (Microsoft 365 LTSC)) [Computer software]. <https://www.microsoft.com>

- Na, W., Ma, B., Shi, S., Chen, Y., Zhang, H., Zhan, Y., & An, H. (2020). Procyanidin B1, a novel and specific inhibitor of Kv10.1 channel, suppresses the evolution of hepatoma. *Biochemical Pharmacology*, 178, 114089. <https://doi.org/10.1016/j.bcp.2020.114089>
- OpenAI (2025): Linguistic revision and structural support for master's thesis writing. Generative AI chat. ChatGPT (GPT-4). <https://chatgpt.com>
- Ouadid-Ahidouch, H., Ahidouch, A., & Pardo, L. A. (2016). Kv10.1 K⁺ channel: From physiology to cancer. *Pflügers Archiv - European Journal of Physiology*, 468(5), 751–762. <https://doi.org/10.1007/s00424-015-1784-3>
- Pardo, L. A., del Camino, D., Sanchez, A., Alves, F., Bruggemann, A., Beckh, S., & Stuhmer, W. (1999). Oncogenic potential of EAG K⁺ channels. *EMBO Journal*, 18(20), 5540–5547. <https://doi.org/10.1093/emboj/18.20.5540>
- Pardo, L. A., & Stuhmer, W. (2014). The roles of K⁺ channels in cancer. *Nature Reviews Cancer*, 14(1), 39–48. <https://doi.org/10.1038/nrc3635>
- Python Software Foundation. (2024). *Python* (Version 3.12.2) [Computer software]. <https://www.python.org>
- Rüenauver, J. (2025). Coarse-grained simulations of the Kv10.1 channel in complex lipid membranes. Master's thesis, University of Vienna, Vienna.
- Rodríguez-Rasgado, J. A., Acuña-Macías, I., & Camacho, J. (2012). Eag1 Channels as Potential Cancer Biomarkers. *Sensors*, 12(5), 5986–5995. <https://doi.org/10.3390/s120505986>
- Saganich, M. J., Machado, E., & Rudy, B. (2001). Differential Expression of Genes Encoding Subthreshold-Operating Voltage-Gated K⁺ Channels in Brain. *The Journal of Neuroscience*, 21(13), 4609–4624. <https://doi.org/10.1523/JNEUROSCI.21-13-04609.2001>
- Schrödinger, LLC. (2015). *The PyMOL Molecular Graphics System* (Version 2.3) [Computer software]. <https://pymol.org>
- Tillman, T. S., & Cascio, M. (2003). Effects of Membrane Lipids on Ion Channel Structure and Function. *Cell Biochemistry and Biophysics*, 38(2), 161–190. <https://doi.org/10.1385/CBB:38:2:161>
- Van Meer, G., Voelker, D. R., & Feigenson, G. W. (2008). Membrane lipids: Where they are and how they behave. *Nature Reviews Molecular Cell Biology*, 9(2), 112–124. <https://doi.org/10.1038/nrm2330>

- Vandenberg, J. I., Perry, M. D., Perrin, M. J., Mann, S. A., Ke, Y., & Hill, A. P. (2012). hERG K⁺ Channels: Structure, Function, and Clinical Significance. *Physiological Reviews*, 92(3), 1393–1478. <https://doi.org/10.1152/physrev.00036.2011>
- Warmke, J. W., & Ganetzky, B. (1994). A family of potassium channel genes related to eag in *Drosophila* and mammals. *Proceedings of the National Academy of Sciences*, 91(8), 3438–3442. <https://doi.org/10.1073/pnas.91.8.3438>
- Whicher, J. R., & MacKinnon, R. (2016). Structure of the voltage-gated K⁺ channel Eag1 reveals an alternative voltage sensing mechanism. *Nature*, 532(7599), 339–343. <https://doi.org/10.1038/nature17185>
- Yellen, G. (2002). The voltage-gated potassium channels and their relatives. *Nature*, 419(6902), 35–42. <https://doi.org/10.1038/nature00978>
- Zagotta, W. N., Olivier, N. B., Black, K. D., Young, E. C., Olson, R., & Gouaux, E. (2003). Structural basis for modulation and agonist specificity of HCN pacemaker channels. *Nature*, 425(6954), 200–205. <https://doi.org/10.1038/nature01922>

Appendix

Zusammenfassung (Deutsch)

Der spannungsgesteuerte Kaliumkanal Kv10.1 wird in über 70% aller menschlichen Tumoren überexprimiert und wird mit Zellproliferation, Angiogenese sowie Tumorzellmigration assoziiert. Obwohl der Ionenkanal als pharmakologisches Ziel von großem Interesse ist, sind die strukturellen Grundlagen seiner Regulation durch Lipide in der Membran unzureichend klar. Durch computergestützte Docking-Studien an der Kryo-EM-Struktur von Kv10.1 (PDB: 5K7L) wurden erste Hinweise auf eine potenzielle Cholesterininteraktionsstelle gefunden. Ob diese Region unter physiologischen Membranbedingungen eine stabile und funktionell relevante Bindungstasche darstellt, wurde bislang nicht untersucht.

Die vorliegende Arbeit untersucht diese Fragestellung mithilfe von molekulardynamischen Simulationen des Kv10.1-Tetramers, eingebettet in eine cholesterinhaltige Lipiddoppelmembran. Zwei unabhängige Simulationen mit einer Länge von jeweils ca. 1,5 μ s wurden analysiert, wobei acht Residuen im Fokus standen.

Zweidimensionale Cholesterindichtekarten der oberen Membranregion zeigten in beiden Simulationen eine reproduzierbare, heterogene Cholesterinverteilung um den Kanal. Die spezifische Kontaktanalyse der Residuen, mit Auswertung von Cholesterin-Belegung, mittlerer Kontaktdauer, simultanen Lipidkontakten, Stabilität der Taschen und Cholesterin-Verweilzeiten, identifizierte Residuum 466 (PHE) als das am häufigsten kontaktierte Residuum über alle vier Untereinheiten und beide Simulationsläufe. Die Residuen 454 (LEU) und 450 (ILE) zeigten mehrere Interaktionen in spezifischen Untereinheiten. Im Gegensatz dazu war Residuum 456 (ALA) unter allen Bedingungen nur minimalistisch involviert.

Zusammenfassend liefern die Simulationen Hinweise auf eine bevorzugte Cholesterin-Interaktionsstelle in der Transmembrandomäne von Kv10.1, die vor allem durch die Residuen 466 (PHE), 454 (LEU) und 450 (ILE) definiert wird. Zudem unterstützen diese Ergebnisse die Hypothese, dass eine Interaktion von Cholesterin an dieser Stelle die Kanalöffnung modulieren könnte und daher relevant im Kontext für die veränderte Membranzusammensetzung in Tumorzellen sein könnte.

Python script: Per-residue cholesterol interaction analysis of Kv10.1 subunits

The following Python script was developed for the per-residue cholesterol interaction analysis of individual Kv10.1 subunits. The script written in Python 3 using MDAnalysis, NumPy, SciPy, pandas, Matplotlib and NetworkX. All analysis was performed using this script on MARTINI CG-MD trajectory data.

```

"""
chol_subunit_analysis.py
=====
Cholesterol interaction analysis for selected pocket residues,
per subunit – Run 1 vs Run 2 comparison.

Selected residues:
    450 (ILE), 453 (LEU), 454 (LEU), 456 (ALA),
    457 (THR), 462 (VAL), 465 (ILE), 466 (PHE)

METRICS PER RESIDUE:
- Occupancy          (% frames with >= 1 CHOL contact)
- Duration           (mean continuous contact event length,  $\mu$ s)
- Lipid Count        (mean CHOL molecules in contact per frame)
- Residence Time     (1/koff from biexponential survival function,  $\mu$ s)
- Stability Score    (consistency of occupancy across time blocks, 0-1)

POCKET STABILITY:
- Trajectory split into N equal time blocks
- Stability Score = 1 - (std / mean) across blocks, clamped [0,1]
    1.0 = CHOL consistently present | 0.0 = highly variable or absent

BINDING SITE LEVEL:
- Occupancy, Duration, Lipid Count, Residence Time, Stability Score

DISTANCE:
- Min distance to nearest CHOL per frame, per residue subplot

COMPARISON (Run 1 vs Run 2):
- Side-by-side bar charts, difference plots, distance overlay,
  stability comparison, binding site comparison, summary heatmap
- Combined CSV with all metrics and delta columns

TIME FACTOR (corrected):
    dt = 2000 ps/frame  $\times$  1e-6 = 0.000002  $\mu$ s/frame
    Total simulation  $\sim$  828 frames  $\times$  0.000002 = 1.656  $\mu$ s

OUTPUT:
    CHOL_{subunit_name}/
    Run1/           per-run plots + CSVs
    Run2/           per-run plots + CSVs
    Comparison/     comparison plots + Comparison_AllMetrics.csv

HOW TO SWITCH SUBUNIT:
    Change the `subunit_name` and `runs` file paths in the Settings section.
    Example:
        subunit_name = "SubA"
        runs = {
            "Run1": {"traj": "r1_subA_chol.xtc", "top": "r1_subA_chol.gro"},
            "Run2": {"traj": "r2_subA_chol.xtc", "top": "r2_subA_chol.gro"},
        }

Requirements:
    pip install MDAnalysis numpy matplotlib scipy pandas networkx
Usage:
    python chol_subunit_analysis.py
"""

import os
import gc
import sys
import time
import logging
import numpy as np
import pandas as pd
import networkx as nx
import matplotlib
matplotlib.use("Agg")

```

```
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.colors as mcolors
from datetime import datetime
from scipy.optimize import curve_fit
import MDAnalysis as mda
from MDAnalysis.analysis.distances import distance_array

# =====
# SETTINGS – edit this section only
# =====

subunit_name = "SubA"          # label used in output folder and plot titles
                                # change to SubB / SubC / SubD as needed

runs = {
    "Run1": {
        "traj": "r1_subA_chol.xtc",
        "top":  "r1_subA_chol.gro",
    },
    "Run2": {
        "traj": "r2_subA_chol.xtc",
        "top":  "r2_subA_chol.gro",
    },
}

# Residues of interest with 3-letter amino acid codes
residues_of_interest = [450, 453, 454, 456, 457, 462, 465, 466]
residue_labels = {
    450: "ILE",
    453: "LEU",
    454: "LEU",
    456: "ALA",
    457: "THR",
    462: "VAL",
    465: "ILE",
    466: "PHE",
}

# CHOL settings (MARTINI 2)
lipid      = "CHOL"
lipid_atom = "ROH"          # headgroup bead
cutoffs    = [0.475, 0.6]   # nm: 4.75 Å inner / 6.0 Å outer dual-cutoff

# Trajectory settings
stride     = 1
dt_traj    = 2000.0          # ps per frame – verified from trajectory

# Analysis settings
binding_site_size = 4        # min residues per binding site
corr_threshold    = 0.3      # Pearson r for binding site detection
n_stability_blocks = 5       # time blocks for stability assessment

# Output settings
timeunit     = "us"          # μs – do not change unless re-verifying dt
fig_format   = "pdf"

# Plot colours
RUN_COLORS   = {"Run1": "#2171b5", "Run2": "#e05c5c"}
RUN_HATCHES  = {"Run1": "",      "Run2": "///"}

# =====
# DERIVED – do not edit below unless you know what you are doing
# =====

cutoff_low_ang = cutoffs[0] * 10.0    # 4.75 Å
cutoff_high_ang = cutoffs[1] * 10.0    # 6.00 Å
```

```
# CORRECTED time factor: ps → μs (2000 ps × 1e-6 = 0.000002 μs/frame)
# Total sim = 828 frames × 0.000002 μs = 1.656 μs ✓
time_factor = 1e-6 if timeunit == "us" else 1e-3
ylabel_t     = r"$\mu$s" if timeunit == "us" else "ns"

output_root = f"CHOL_{subunit_name}"
comp_dir    = os.path.join(output_root, "Comparison")
for d in [comp_dir] + [os.path.join(output_root, r) for r in runs]:
    os.makedirs(d, exist_ok=True)

def fout(run_name, fname):
    return os.path.join(output_root, run_name, fname)

def fcomp(fname):
    return os.path.join(comp_dir, fname)

def res_label(resid):
    """Return 'resid (AAA)' - e.g. '466 (PHE)'."""
    aa = residue_labels.get(resid, "")
    return f"{resid} ({aa})" if aa else str(resid)

# =====
# LOGGING
# =====

log_filename = f"chol_{subunit_name.lower()}_rlvr2_" \
    f"{datetime.now().strftime('%Y%m%d_%H%M%S')}.log"
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s | %(levelname)s | %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S",
    handlers=[
        logging.FileHandler(log_filename),
        logging.StreamHandler(sys.stdout),
    ],
)
logger = logging.getLogger(__name__)

def log_step(n, desc):
    logger.info("")
    logger.info("=" * 60)
    logger.info(f"  STEP {n}: {desc}")
    logger.info("=" * 60)

def elapsed(t0):
    s = time.time() - t0
    return f"{s:.1f} s" if s < 60 else f"{s/60:.1f} min"

# =====
# HELPER FUNCTIONS
# =====

def get_durations(series, dt):
    """List of continuous contact event lengths (time units)."""
    d = np.diff(np.concatenate([[0], series.astype(int), [0]]))
    starts = np.where(d == 1)[0]
    ends = np.where(d == -1)[0]
    return list((ends - starts) * dt)

def fit_residence_time(series, dt):
    """
    Biexponential survival function fit → (residence_time, r2).
    RT = 1/koff (slow component). Returns (nan, nan) if insufficient data.
    NOTE: RT may exceed simulation length when dissociation is not observed -
    this is a lower-bound estimate, confirming near-irreversible binding.
    """
    idx = np.where(series)[0]
```

```
if len(idx) < 5:
    return np.nan, np.nan
n = len(series)
max_lag = max(1, n // 2)
sigma = np.zeros(max_lag)
for lag in range(max_lag):
    valid = idx[idx + lag < n]
    if len(valid) == 0:
        break
    sigma[lag] = series[valid + lag].sum() / len(idx)
t_arr = np.arange(max_lag) * dt
norm = sigma / (sigma[0] + 1e-12)

def biexp(t, A1, k1, A2, k2):
    return A1 * np.exp(-k1 * t) + A2 * np.exp(-k2 * t)
try:
    popt, _ = curve_fit(
        biexp, t_arr, norm,
        p0=[0.5, 1.0, 0.5, 0.1],
        bounds=([0, 0, 0, 0], [1, np.inf, 1, np.inf]),
        maxfev=20000)
    koff = min(popt[1], po[3])
    rt = 1.0 / (koff + 1e-12)
    fitted = biexp(t_arr, *popt)
    ss_res = np.sum((norm - fitted) ** 2)
    ss_tot = np.sum((norm - norm.mean()) ** 2)
    r2 = 1.0 - ss_res / (ss_tot + 1e-12)
    return round(rt, 6), round(r2, 4)
except Exception:
    return np.nan, np.nan

def compute_stability(any_contact_mat, n_blocks, dt_t):
    """
    Split trajectory into n_blocks. Compute occupancy per block per residue.
    Stability Score = 1 - (std / (mean + eps)), clamped [0, 1].
    Residues with mean occupancy < 1% → score = 0.
    Returns: block_occs (n_blocks, n_res), stability (n_res,), block_times (n_blocks,)
    """
    n_frames, n_res = any_contact_mat.shape
    block_size = n_frames // n_blocks
    block_occs = np.zeros((n_blocks, n_res))
    block_times = np.zeros(n_blocks)
    for b in range(n_blocks):
        start = b * block_size
        end = (b + 1) * block_size if b < n_blocks - 1 else n_frames
        block_occs[b, :] = any_contact_mat[start:end, :].mean(axis=0) * 100.0
        block_times[b] = ((start + end) / 2.0) * dt_t
    mean_occ = block_occs.mean(axis=0)
    std_occ = block_occs.std(axis=0)
    stability = np.clip(1.0 - std_occ / (mean_occ + 1e-6), 0.0, 1.0)
    stability[mean_occ < 1.0] = 0.0
    return block_occs, stability, block_times

# =====
# CORE PER-RUN ANALYSIS
# =====

def run_analysis(run_name, traj_file, top_file):
    logger.info(f"\n{'#' * 60}\n {run_name} | {subunit_name}\n "
               f"Traj: {traj_file} Top: {top_file}\n{'#' * 60}")

    # Load universe
    t = time.time()
    u = mda.Universe(top_file, traj_file)
    n_frames_total = len(u.trajectory)
    logger.info(f"[{run_name}] {u.atoms.n_atoms} atoms | "
               f"{n_frames_total} frames [{elapsed(t)}]")
```

```

# Selections
chol_sel = u.select_atoms(f"resname {lipid} and name {lipid_atom}")
n_chol = len(chol_sel)

sel_resids = []
res_selections = {}
for resid in residues_of_interest:
    sel = u.select_atoms(f"resid {resid}")
    if len(sel) == 0:
        logger.warning(f"[{run_name}] Residue {resid} not found - skipping")
        continue
    res_selections[resid] = sel
    sel_resids.append(resid)
n_res = len(sel_resids)

logger.info(f"[{run_name}] Residues : {[res_label(r) for r in sel_resids]}")
logger.info(f"[{run_name}] CHOL ROH : {n_chol} [{elapsed(t)}]")

if n_res == 0:
    raise ValueError(f"[{run_name}] No residues found.")
if n_chol == 0:
    raise ValueError(f"[{run_name}] No {lipid} {lipid_atom} beads found.")

# Time axis
dt_ps = dt_traj * stride
dt_t = dt_ps * time_factor # μs per frame
frames_to = list(range(0, n_frames_total, stride))
n_frames = len(frames_to)
time_axis = np.arange(n_frames) * dt_t
logger.info(f"[{run_name}] dt = {dt_ps:.0f} ps = {dt_t:.6f} {timeunit}/frame | "
            f"n_frames = {n_frames} | "
            f"total = {time_axis[-1]:.4f} {timeunit}")

# Contact collection - dual cutoff
t = time.time()
logger.info(f"[{run_name}] Collecting contacts ...")
contact_series = np.zeros((n_frames, n_res, n_chol), dtype=np.int8)
dist_matrix = np.full((n_frames, n_res), np.nan)
in_contact = np.zeros((n_res, n_chol), dtype=bool)

for fi, ts_idx in enumerate(frames_to):
    u.trajectory[ts_idx]
    chol_pos = chol_sel.positions
    for ri, resid in enumerate(sel_resids):
        res_pos = res_selections[resid].positions
        dists = distance_array(res_pos, chol_pos, box=u.dimensions)
        min_d = dists.min(axis=0)
        enters = (~in_contact[ri]) & (min_d <= cutoff_low_ang)
        exits = in_contact[ri] & (min_d > cutoff_high_ang)
        in_contact[ri, enters] = True
        in_contact[ri, exits] = False
        contact_series[fi, ri, :] = in_contact[ri].astype(np.int8)
        dist_matrix[fi, ri] = min_d.min()
    if fi % 100 == 0 or fi == n_frames - 1:
        logger.info(f"[{run_name}] Frame {fi+1}/{n_frames} [{elapsed(t)}]")

logger.info(f"[{run_name}] Contact collection done [{elapsed(t)}]")

# Safe close - avoid _xdr AttributeError
try:
    u.trajectory.close()
except Exception:
    pass
try:
    del u
except Exception:
    pass

```

```

gc.collect()

any_contact_mat = contact_series.any(axis=2).astype(float)

# Per-residue metrics
t = time.time()
res_occupancy = np.zeros(n_res)
res_duration = np.zeros(n_res)
res_lipidcount = np.zeros(n_res)
res_restime = np.full(n_res, np.nan)
res_r2 = np.full(n_res, np.nan)

for ri, resid in enumerate(sel_resids):
    cs = contact_series[:, ri, :]
    any_c = any_contact_mat[:, ri]
    res_occupancy[ri] = any_c.mean() * 100.0
    res_lipidcount[ri] = cs.sum(axis=1).mean()
    all_dur = []
    for ci in range(n_chol):
        all_dur.extend(get_durations(cs[:, ci], dt_t))
    res_duration[ri] = np.mean(all_dur) if all_dur else 0.0
    rt, r2 = fit_residence_time(any_c.astype(int), dt_t)
    res_restime[ri] = rt
    res_r2[ri] = r2
    logger.info(f"[{run_name}] {res_label(resid)}: "
               f"Occ={res_occupancy[ri]:.1f}% "
               f"Dur={res_duration[ri]:.6f} {timeunit} "
               f"LC={res_lipidcount[ri]:.2f} "
               f"RT={rt} {timeunit}")

logger.info(f"[{run_name}] Residue metrics done [{elapsed(t)}]")

# Pocket stability
t = time.time()
logger.info(f"[{run_name}] Computing pocket stability ({n_stability_blocks}
blocks)...")
block_occs, stability_score, block_times = compute_stability(
    any_contact_mat, n_stability_blocks, dt_t)
for ri, resid in enumerate(sel_resids):
    bo = [round(block_occs[b, ri], 1) for b in range(n_stability_blocks)]
    logger.info(f"[{run_name}] {res_label(resid)} | "
               f"Stability={stability_score[ri]:.3f} | "
               f"Block occs={bo}")
logger.info(f"[{run_name}] Stability done [{elapsed(t)}]")

# Binding site detection
t = time.time()
corrcoef = np.nan_to_num(np.corrcoef(any_contact_mat.T), nan=0.0)
G = nx.Graph()
G.add_nodes_from(range(n_res))
for i in range(n_res):
    for j in range(i + 1, n_res):
        if corrcoef[i, j] >= corr_threshold:
            G.add_edge(i, j, weight=corrcoef[i, j])
components = sorted(
    [c for c in nx.connected_components(G) if len(c) >= binding_site_size],
    key=lambda c: -len(c))
node_list = [sorted(list(c)) for c in components]

if len(node_list) == 0:
    logger.warning(f"[{run_name}] No binding sites detected "
                  f"(corr_threshold={corr_threshold})")
else:
    logger.info(f"[{run_name}] {len(node_list)} binding site(s):")
    for bs_id, nodes in enumerate(node_list):
        logger.info(f"  BS{bs_id}: {[res_label(sel_resids[i]) for i in nodes]}")

# Binding site metrics

```

```

bs_ids      = []
bs_resids_all = []
bs_occupancy = []
bs_duration = []
bs_lipidcount = []
bs_restime = []
bs_r2       = []
bs_stability = []
bs_id_per_res = np.full(n_res, -1)

for bs_id, nodes in enumerate(node_list):
    bs_cs      = contact_series[:, nodes, :]
    any_chol_bs = bs_cs.any(axis=1).any(axis=1)
    chol_in_bs  = bs_cs.any(axis=1)
    dur_list    = []
    for ci in range(n_chol):
        dur_list.extend(get_durations(chol_in_bs[:, ci].astype(int), dt_t))
    rt_bs, r2_bs = fit_residence_time(any_chol_bs.astype(int), dt_t)
    mean_stab    = float(stability_score[list(nodes)].mean())
    bs_ids.append(bs_id)
    bs_resids_all.append([sel_resids[i] for i in nodes])
    bs_occupancy.append(round(any_chol_bs.mean() * 100.0, 3))
    bs_duration.append(round(np.mean(dur_list) if dur_list else 0.0, 8))
    bs_lipidcount.append(round(chol_in_bs.sum(axis=1).mean(), 4))
    bs_restime.append(rt_bs)
    bs_r2.append(r2_bs)
    bs_stability.append(round(mean_stab, 4))
    for ni in nodes:
        bs_id_per_res[ni] = bs_id
    logger.info(f"[{run_name}] BS{bs_id}: "
                f"Occ={bs_occupancy[-1]:.1f}% "
                f"Dur={bs_duration[-1]:.6f} "
                f"LC={bs_lipidcount[-1]:.2f} "
                f"RT={rt_bs} "
                f"Stability={mean_stab:.3f}")

# DataFrames + CSVs
df_res = pd.DataFrame({
    "Residue":      [res_label(r) for r in sel_resids],
    "Residue ID":   sel_resids,
    "Occupancy (%)": np.round(res_occupancy, 3),
    f"Duration ({timeunit})": np.round(res_duration, 8),
    "Lipid Count":  np.round(res_lipidcount, 4),
    f"Residence Time ({timeunit})": np.round(res_restime, 6),
    "ResTime R2":   np.round(res_r2, 4),
    "Stability Score": np.round(stability_score, 4),
    "Binding Site ID": bs_id_per_res,
})
df_bs = pd.DataFrame({
    "Binding Site ID": bs_ids,
    "Residues":        [str(r) for r in bs_resids_all],
    "Size":            [len(r) for r in bs_resids_all],
    "Occupancy (%)":   np.round(bs_occupancy, 3),
    f"Duration ({timeunit})": np.round(bs_duration, 8),
    "Lipid Count":     np.round(bs_lipidcount, 4),
    f"Residence Time ({timeunit})": bs_restime,
    "ResTime R2":      bs_r2,
    "Stability Score": np.round(bs_stability, 4),
}).sort_values("Occupancy (%)", ascending=False).reset_index(drop=True)
df_stab = pd.DataFrame(
    block_occs, columns=[res_label(r) for r in sel_resids])
df_stab.insert(0, f"Block_midtime_{timeunit}", block_times)
df_dist = pd.DataFrame(
    dist_matrix, columns=[res_label(r) for r in sel_resids])
df_dist.insert(0, f"Time_{timeunit}", time_axis)

df_res.to_csv(fout(run_name, "ResidueMetrics.csv"), index=False)
df_bs.to_csv(fout(run_name, "BindingSiteInfo.csv"), index=False)

```

```
df_stab.to_csv(fout(run_name, "StabilityBlocks.csv"), index=False)
df_dist.to_csv(fout(run_name, "DistanceMatrix.csv"), index=False)
logger.info(f"[{run_name}] CSVs saved")

# Per-run plots
_plot_individual(run_name, sel_resids, n_res,
                 res_occupancy, res_duration, res_lipidcount,
                 res_restime, stability_score,
                 block_occs, block_times,
                 dist_matrix, time_axis,
                 any_contact_mat, corrcoef,
                 bs_ids, bs_occupancy, bs_duration,
                 bs_lipidcount, bs_restime, bs_stability)

return {
    "sel_resids": sel_resids,
    "n_res": n_res,
    "time_axis": time_axis,
    "dt_t": dt_t,
    "res_occupancy": res_occupancy,
    "res_duration": res_duration,
    "res_lipidcount": res_lipidcount,
    "res_restime": res_restime,
    "stability_score": stability_score,
    "block_occs": block_occs,
    "block_times": block_times,
    "dist_matrix": dist_matrix,
    "any_contact_mat": any_contact_mat,
    "corrcoef": corrcoef,
    "node_list": node_list,
    "bs_ids": bs_ids,
    "bs_resids_all": bs_resids_all,
    "bs_occupancy": bs_occupancy,
    "bs_duration": bs_duration,
    "bs_lipidcount": bs_lipidcount,
    "bs_restime": bs_restime,
    "bs_stability": bs_stability,
    "df_res": df_res,
    "df_bs": df_bs,
}

# =====
# PER-RUN INDIVIDUAL PLOTS
# =====

def _plot_individual(run_name, sel_resids, n_res,
                    res_occupancy, res_duration, res_lipidcount,
                    res_restime, stability_score,
                    block_occs, block_times,
                    dist_matrix, time_axis,
                    any_contact_mat, corrcoef,
                    bs_ids, bs_occupancy, bs_duration,
                    bs_lipidcount, bs_restime, bs_stability):

    color = RUN_COLORS[run_name]
    x = np.arange(n_res)
    fw = max(8, n_res * 1.1)
    xlabels = [res_label(r) for r in sel_resids]

    def xticks(ax, rot=45):
        ax.set_xticks(x)
        ax.set_xticklabels(xlabel, rotation=rot, ha="right", fontsize=10)

    # — Bar + logo for each residue metric —————
    metric_cfg = [
        ("Occupancy_pct",
         res_occupancy,
         "Occupancy (%)",
```

```

        "CHOL Occupancy per Residue"),
    ("Duration",
     res_duration,
     f"Duration ({ylabel_t})",
     "Mean CHOL Contact Duration per Residue"),
    ("Lipid_Count",
     res_lipidcount,
     "Lipid Count",
     "Mean Simultaneous CHOL Contacts per Residue"),
    ("Residence_Time",
     np.nan_to_num(res_restime, nan=0),
     f"Residence Time ({ylabel_t})",
     "CHOL Residence Time per Residue"),
    ("Stability_Score",
     stability_score,
     "Stability Score (0-1)",
     "Pocket Stability Score per Residue"),
]

for fname_key, values, ylabel, title in metric_cfg:
    # bar plot
    fig, ax = plt.subplots(figsize=(fw, 4.5))
    ax.bar(x, values, color=color, edgecolor="white", linewidth=0.5)
    xticks(ax)
    ax.set_xlabel("Residue", fontsize=11)
    ax.set_ylabel(ylabel, fontsize=11)
    ax.set_title(f"{title}\n{run_name} | {subunit_name}", fontsize=11)
    ax.grid(axis="y", alpha=0.25, linewidth=0.5)
    plt.tight_layout()
    plt.savefig(fout(run_name, f"CHOL_{fname_key}.{fig_format}"), dpi=200)
    plt.close()

    # logo plot (sorted descending)
    order = np.argsort(values)[::-1]
    fig, ax = plt.subplots(figsize=(fw, 4.5))
    ax.vlines(x, 0, values[order], color=color, linewidth=2.0, alpha=0.7)
    ax.scatter(x, values[order], color=color, s=60, zorder=3)
    ax.set_xticks(x)
    ax.set_xticklabels(
        [res_label(r) for r in np.array(sel_resids)[order]],
        rotation=45, ha="right", fontsize=10)
    ax.set_xlabel("Residue (ranked by value)", fontsize=11)
    ax.set_ylabel(ylabel, fontsize=11)
    ax.set_title(f"{title} - Ranked\n{run_name} | {subunit_name}", fontsize=11)
    ax.grid(axis="y", alpha=0.25, linewidth=0.5)
    plt.tight_layout()
    plt.savefig(fout(run_name, f"CHOL_{fname_key}_logo.{fig_format}"), dpi=200)
    plt.close()

logger.info(f"[{run_name}] Metric bar + logo plots saved")

# — Stability: block occupancy line plot —————
cmap_s = plt.cm.get_cmap("RdYlGn")
fig, ax = plt.subplots(figsize=(fw, 4.5))
for b in range(len(block_times)):
    ax.plot(x, block_occs[b, :],
           color=cmap_s(b / max(1, len(block_times) - 1)),
           linewidth=1.8, alpha=0.85,
           label=f"Block {b+1} ({block_times[b]:.3f} {timeunit})")
xticks(ax)
ax.set_xlabel("Residue", fontsize=11)
ax.set_ylabel("Occupancy (%) per time block", fontsize=11)
ax.set_title(
    f"Cholesterol Occupancy per Time Block\n"
    f"{run_name} | {subunit_name} | "
    f"Overlapping lines = stable interaction",
    fontsize=11)
ax.legend(fontsize=9, loc="upper right")

```

```

ax.grid(axis="y", alpha=0.25, linewidth=0.5)
plt.tight_layout()
plt.savefig(fout(run_name, f"CHOL_Stability_blocks.{fig_format}"), dpi=200)
plt.close()
logger.info(f"[{run_name}] Stability block plot saved")

# — Stability: heatmap (blocks × residues) —————
fig, ax = plt.subplots(figsize=(fw, max(3.0, n_stability_blocks * 0.7)))
im = ax.imshow(block_occs, aspect="auto", cmap="YlOrRd",
               vmin=0, vmax=100,
               extent=[-0.5, n_res - 0.5,
                       len(block_times) - 0.5, -0.5])
xticks(ax)
ax.set_yticks(np.arange(len(block_times)))
ax.set_yticklabels([f"Block {b+1} ({block_times[b]:.3f} {timeunit})"
                    for b in range(len(block_times))], fontsize=9)
ax.set_xlabel("Residue", fontsize=11)
ax.set_title(
    f"Cholesterol Occupancy Heatmap per Time Block\n"
    f"{run_name} | {subunit_name} | Uniform rows = stable pocket",
    fontsize=11)
fig.colorbar(im, ax=ax, label="Occupancy (%)", shrink=0.8)
plt.tight_layout()
plt.savefig(fout(run_name, f"CHOL_Stability_heatmap.{fig_format}"),
            dpi=200, bbox_inches="tight")
plt.close()
logger.info(f"[{run_name}] Stability heatmap saved")

# — Contact heatmap (residues × time) —————
binary = any_contact_mat.T
fig, ax = plt.subplots(figsize=(12, max(3, n_res * 0.7)))
cmap_b = mcolors.ListedColormap(["#f0f0f0", color])
ax.imshow(binary, aspect="auto", cmap=cmap_b, vmin=0, vmax=1,
          interpolation="nearest",
          extent=[time_axis[0], time_axis[-1], n_res - 0.5, -0.5])
ax.set_yticks(x)
ax.set_yticklabels(xlabels, fontsize=10)
ax.set_xlabel(f"Simulation time ({ylabel_t})", fontsize=11)
ax.set_ylabel("Residue", fontsize=11)
ax.set_title(
    f"Cholesterol Contact Timeline\n"
    f"{run_name} | {subunit_name} | "
    f"Coloured = within {cutoff_high_ang:.0f} Å",
    fontsize=11)
plt.tight_layout()
plt.savefig(fout(run_name, f"CHOL_contact_heatmap.{fig_format}"),
            dpi=200, bbox_inches="tight")
plt.close()
logger.info(f"[{run_name}] Contact heatmap saved")

# — Correlation matrix —————
sz = max(5, n_res * 0.9)
fig, ax = plt.subplots(figsize=(sz, sz))
im = ax.imshow(corrcoef, aspect="auto", cmap="RdBu_r", vmin=-1, vmax=1)
ax.set_xticks(x)
ax.set_xticklabels(xlabels, rotation=45, ha="right", fontsize=10)
ax.set_yticks(x)
ax.set_yticklabels(xlabels, fontsize=10)
ax.set_title(
    f"CHOL Contact Correlation Matrix\n"
    f"{run_name} | {subunit_name}",
    fontsize=11)
fig.colorbar(im, ax=ax, label="Pearson r", shrink=0.8)
plt.tight_layout()
plt.savefig(fout(run_name, f"CHOL_CorrCoef.{fig_format}"),
            dpi=200, bbox_inches="tight")
plt.close()

```

```

logger.info(f"[{run_name}] Correlation matrix saved")

# — Distance over time subplots —————
n_cols = 4
n_rows = int(np.ceil(n_res / n_cols))
fig, axes = plt.subplots(n_rows, n_cols,
                        figsize=(n_cols * 4.0, n_rows * 3.2),
                        sharex=True)
axes_flat = axes.flatten() if n_res > 1 else [axes]
for ri, resid in enumerate(sel_resids):
    ax = axes_flat[ri]
    d = dist_matrix[:, ri]
    ic = d <= cutoff_high_ang
    ax.axhline(cutoff_high_ang, color="#e05c5c",
              linewidth=0.9, linestyle="--", alpha=0.8)
    ax.fill_between(time_axis, 0, cutoff_high_ang,
                  alpha=0.07, color="#e05c5c")
    ax.plot(time_axis, d, color="#bbbbbb", linewidth=0.6, alpha=0.8)
    ax.scatter(time_axis[ic], d[ic], s=3, color=color, zorder=3)
    ax.set_title(
        f"{res_label(resid)}\nOcc: {res_occupancy[ri]:.1f}%",
        fontsize=9, pad=3)
    ax.set_ylim(0, max(float(np.nanmax(d)) * 1.1, cutoff_high_ang * 2))
    ax.yaxis.set_major_locator(ticker.MaxNLocator(3))
    ax.tick_params(labelsize=7)
for ax in axes_flat[n_res:]:
    ax.set_visible(False)
for ax in axes_flat[(n_rows - 1) * n_cols: n_res]:
    ax.set_xlabel(f"Time ({ylabel_t})", fontsize=9)
for ri in range(0, n_res, n_cols):
    axes_flat[ri].set_ylabel("Min dist (Å)", fontsize=9)
handles = [
    plt.Line2D([0],[0], color="#e05c5c", linestyle="--",
              linewidth=1, label=f"{cutoff_high_ang:.0f} Å cutoff"),
    plt.Line2D([0],[0], marker="o", color="w",
              markerfacecolor=color, markersize=6,
              label=f"≤ {cutoff_high_ang:.0f} Å"),
]
fig.legend(handles=handles, loc="upper right", fontsize=9,
          framealpha=0.85, ncol=2)
fig.suptitle(
    f"Minimum Distance to Nearest CHOL per Residue\n"
    f"{run_name} | {subunit_name}",
    fontsize=11, y=1.01)
plt.tight_layout()
plt.savefig(fout(run_name, f"CHOL_distance_over_time.{fig_format}"),
          dpi=150, bbox_inches="tight")
plt.close()
logger.info(f"[{run_name}] Distance over time saved")

# — Binding site scatter plots —————
if len(bs_ids) > 0:
    bs_cfg = [
        ("Occupancy", np.array(bs_occupancy), "Occupancy (%)",
        ("Duration", np.array(bs_duration), f"Duration ({ylabel_t})",
        ("Lipid_Count", np.array(bs_lipidcount), "Lipid Count"),
        ("Residence_Time",
         np.array([v if not (isinstance(v, float) and np.isnan(v))
                    else 0 for v in bs_restime]),
         f"Residence Time ({ylabel_t}")),
        ("Stability", np.array(bs_stability), "Stability Score (0-1)"),
    ]
    bs_titles = {
        "Occupancy": "Binding Site CHOL Occupancy",
        "Duration": "Binding Site Mean Contact Duration",
        "Lipid_Count": "Binding Site Mean Lipid Count",
        "Residence_Time": "Binding Site CHOL Residence Time",
        "Stability": "Binding Site Pocket Stability Score",
    }

```

```

    }
    for name, values, ylabel in bs_cfg:
        order = np.argsort(values)[::-1]
        fig, ax = plt.subplots(figsize=(max(4, len(bs_ids) * 1.0), 4))
        ax.scatter(np.arange(len(values)), values[order], s=90, color=color)
        for i, (bid, val) in enumerate(
            zip(np.array(bs_ids)[order], values[order])):
            ax.annotate(f"BS{int(bid)}", (i, val),
                        textcoords="offset points",
                        xytext=(0, 7), fontsize=9, ha="center")
        ax.set_xticks(np.arange(len(bs_ids)))
        ax.set_xticklabels([f"BS{int(b)}" for b in np.array(bs_ids)[order]])
        ax.set_xlabel("Binding Site", fontsize=11)
        ax.set_ylabel(ylabel, fontsize=11)
        ax.set_title(
            f"{bs_titles[name]}\n{run_name} | {subunit_name}",
            fontsize=11)
        ax.grid(axis="y", alpha=0.3)
        plt.tight_layout()
        plt.savefig(fout(run_name,
                        f"CHOL_{name}_v_binding_site.{fig_format}"), dpi=200)
        plt.close()
    logger.info(f"[{run_name}] Binding site plots saved")

# =====
# COMPARISON PLOT FUNCTIONS
# =====

def plot_comparison_bar(sel_resids, v1, v2, ylabel, title, fname):
    x, w = np.arange(len(sel_resids)), 0.38
    xlabels = [res_label(r) for r in sel_resids]
    fw = max(8, len(sel_resids) * 1.1)
    fig, ax = plt.subplots(figsize=(fw, 4.5))
    ax.bar(x - w/2, v1, w, label="Run 1",
           color=RUN_COLORS["Run1"], edgecolor="white", linewidth=0.4)
    ax.bar(x + w/2, v2, w, label="Run 2",
           color=RUN_COLORS["Run2"], edgecolor="white",
           linewidth=0.4, hatch=RUN_HATCHES["Run2"])
    ax.set_xticks(x)
    ax.set_xticklabels(xlabel, rotation=45, ha="right", fontsize=10)
    ax.set_xlabel("Residue", fontsize=11)
    ax.set_ylabel(ylabel, fontsize=11)
    ax.set_title(f"{title}\nRun 1 vs Run 2 | {subunit_name}", fontsize=11)
    ax.legend(fontsize=10)
    ax.grid(axis="y", alpha=0.25, linewidth=0.5)
    plt.tight_layout()
    plt.savefig(fname, dpi=200)
    plt.close()
    logger.info(f" Saved: {fname}")

def plot_comparison_diff(sel_resids, v1, v2, ylabel, title, fname):
    diff = np.array(v2, dtype=float) - np.array(v1, dtype=float)
    colors = ["#2171b5" if d >= 0 else "#e05c5c" for d in diff]
    xlabels = [res_label(r) for r in sel_resids]
    fw = max(8, len(sel_resids) * 1.1)
    fig, ax = plt.subplots(figsize=(fw, 4.5))
    ax.bar(np.arange(len(sel_resids)), diff,
           color=colors, edgecolor="white", linewidth=0.4)
    ax.axhline(0, color="black", linewidth=0.9)
    ax.set_xticks(np.arange(len(sel_resids)))
    ax.set_xticklabels(xlabel, rotation=45, ha="right", fontsize=10)
    ax.set_xlabel("Residue", fontsize=11)
    ax.set_ylabel(f"Δ {ylabel} (Run2 - Run1)", fontsize=11)
    ax.set_title(
        f"{title} - Difference (Run2 - Run1)\n"
        f"{subunit_name} | Blue = Run2 higher | Red = Run1 higher",
        fontsize=11)

```

```

ax.grid(axis="y", alpha=0.25, linewidth=0.5)
plt.tight_layout()
plt.savefig(fname, dpi=200)
plt.close()
logger.info(f"  Saved: {fname}")

def plot_comparison_distance(sel_resids, dist1, dist2,
                           time1, time2, occ1, occ2, fname):
    n_res = len(sel_resids)
    n_cols = 4
    n_rows = int(np.ceil(n_res / n_cols))
    fig, axes = plt.subplots(n_rows, n_cols,
                             figsize=(n_cols * 4.0, n_rows * 3.2),
                             sharex=False)
    axes_flat = axes.flatten() if n_res > 1 else [axes]
    for ri, resid in enumerate(sel_resids):
        ax = axes_flat[ri]
        d1, d2 = dist1[:, ri], dist2[:, ri]
        ax.axhline(cutoff_high_ang, color="black",
                  linewidth=0.7, linestyle="--", alpha=0.4)
        ax.fill_between(time1, 0, cutoff_high_ang,
                        alpha=0.04, color="grey")
        ax.plot(time1, d1, color=RUN_COLORS["Run1"],
                linewidth=0.8, alpha=0.75, label="Run 1")
        ax.plot(time2, d2, color=RUN_COLORS["Run2"],
                linewidth=0.8, alpha=0.75, label="Run 2")
        ax.set_title(
            f"{res_label(resid)}\nR1: {occ1[ri]:.1f}%  R2: {occ2[ri]:.1f}%",
            fontsize=9, pad=2)
        all_d = np.concatenate([d1, d2])
        ax.set_ylim(0, max(float(np.nanmax(all_d)) * 1.1, cutoff_high_ang * 2))
        ax.yaxis.set_major_locator(ticker.MaxNLocator(3))
        ax.tick_params(labelsize=7)
    for ax in axes_flat[n_res:]:
        ax.set_visible(False)
    for ax in axes_flat[(n_rows - 1) * n_cols: n_res]:
        ax.set_xlabel(f"Time ({ylabel_t})", fontsize=8)
    for ri in range(0, n_res, n_cols):
        axes_flat[ri].set_ylabel("Min dist (Å)", fontsize=8)
    handles = [
        plt.Line2D([0],[0], color=RUN_COLORS["Run1"], linewidth=1.5, label="Run 1"),
        plt.Line2D([0],[0], color=RUN_COLORS["Run2"], linewidth=1.5, label="Run 2"),
        plt.Line2D([0],[0], color="black", linestyle="--",
                  linewidth=1, label=f"{cutoff_high_ang:.0f} Å cutoff"),
    ]
    fig.legend(handles=handles, loc="upper right",
              fontsize=9, framealpha=0.85, ncol=3)
    fig.suptitle(
        f"Minimum Distance to Nearest CHOL – Run 1 vs Run 2\n{n{subunit_name}}",
        fontsize=11, y=1.01)
    plt.tight_layout()
    plt.savefig(fname, dpi=150, bbox_inches="tight")
    plt.close()
    logger.info(f"  Saved: {fname}")

def plot_stability_comparison(sel_resids, r1, r2,
                             fname_bar, fname_block, fname_heatmap):
    x = np.arange(len(sel_resids))
    w = 0.38
    fw = max(8, len(sel_resids) * 1.1)
    xlabels = [res_label(r) for r in sel_resids]
    n_b = n_stability_blocks

    # 1. Stability score bar
    fig, ax = plt.subplots(figsize=(fw, 4.5))
    ax.bar(x - w/2, r1["stability_score"], w, label="Run 1",

```

```

        color=RUN_COLORS["Run1"], edgecolor="white", linewidth=0.4)
ax.bar(x + w/2, r2["stability_score"], w, label="Run 2",
        color=RUN_COLORS["Run2"], edgecolor="white",
        linewidth=0.4, hatch=RUN_HATCHES["Run2"])
ax.axhline(0.5, color="grey", linestyle="--",
           linewidth=0.8, alpha=0.7, label="0.5 reference")
ax.set_xticks(x)
ax.set_xticklabels(xlabels, rotation=45, ha="right", fontsize=10)
ax.set_xlabel("Residue", fontsize=11)
ax.set_ylabel("Stability Score (0-1)", fontsize=11)
ax.set_title(
    f"Pocket Stability Score – Run 1 vs Run 2\n"
    f"{subunit_name} | 1.0 = consistently occupied throughout simulation",
    fontsize=11)
ax.legend(fontsize=9)
ax.grid(axis="y", alpha=0.25, linewidth=0.5)
plt.tight_layout()
plt.savefig(fname_bar, dpi=200)
plt.close()
logger.info(f"  Saved: {fname_bar}")

# 2. Block occupancy overlay
cmap_r1 = plt.cm.get_cmap("Blues")
cmap_r2 = plt.cm.get_cmap("Reds")
fig, ax = plt.subplots(figsize=(fw, 4.5))
for b in range(n_b):
    frac = 0.4 + 0.6 * b / max(1, n_b - 1)
    ax.plot(x, r1["block_occs"][b, :],
            color=cmap_r1(frac), linewidth=1.4, alpha=0.85,
            label=f"R1 B{b+1} ({r1['block_times'][b]:.3f} {timeunit})")
    ax.plot(x, r2["block_occs"][b, :],
            color=cmap_r2(frac), linewidth=1.4, alpha=0.85,
            linestyle="--",
            label=f"R2 B{b+1} ({r2['block_times'][b]:.3f} {timeunit})")
ax.set_xticks(x)
ax.set_xticklabels(xlabels, rotation=45, ha="right", fontsize=10)
ax.set_xlabel("Residue", fontsize=11)
ax.set_ylabel("Occupancy (%) per time block", fontsize=11)
ax.set_title(
    f"Cholesterol Occupancy per Time Block – Run 1 (solid) vs Run 2 (dashed)\n"
    f"{subunit_name} | Overlapping lines = stable interaction",
    fontsize=11)
ax.legend(fontsize=7, loc="upper right", ncol=2)
ax.grid(axis="y", alpha=0.25, linewidth=0.5)
plt.tight_layout()
plt.savefig(fname_block, dpi=200)
plt.close()
logger.info(f"  Saved: {fname_block}")

# 3. Stability heatmap (interleaved Run1/Run2 rows)
combined = np.zeros((n_b * 2, len(sel_resids)))
ytlabels = []
for b in range(n_b):
    combined[b * 2, :] = r1["block_occs"][b, :]
    combined[b * 2 + 1, :] = r2["block_occs"][b, :]
    ytlabels.append(f"R1 B{b+1} ({r1['block_times'][b]:.3f} {timeunit})")
    ytlabels.append(f"R2 B{b+1} ({r2['block_times'][b]:.3f} {timeunit})")
fig, ax = plt.subplots(figsize=(fw, max(4, n_b * 1.0)))
im = ax.imshow(combined, aspect="auto", cmap="YlOrRd",
               vmin=0, vmax=100,
               extent=[-0.5, len(sel_resids) - 0.5,
                       n_b * 2 - 0.5, -0.5])
ax.set_xticks(np.arange(len(sel_resids)))
ax.set_xticklabels(xlabels, rotation=45, ha="right", fontsize=10)
ax.set_yticks(np.arange(n_b * 2))
ax.set_yticklabels(ytlabels, fontsize=8)
ax.set_xlabel("Residue", fontsize=11)
ax.set_title(

```

```

        f"Cholesterol Occupancy per Time Block – Run 1 vs Run 2\n"
        f"{subunit_name} | Uniform rows = stable pocket",
        fontsize=11)
fig.colorbar(im, ax=ax, label="Occupancy (%)", shrink=0.7)
plt.tight_layout()
plt.savefig(fname_heatmap, dpi=200, bbox_inches="tight")
plt.close()
logger.info(f"  Saved: {fname_heatmap}")

def plot_bs_comparison(r1, r2):
    bs_cfg = [
        ("Occupancy", "bs_occupancy", "Occupancy (%)",
         "Binding Site CHOL Occupancy"),
        ("Duration", "bs_duration", f"Duration ({ylabel_t})",
         "Binding Site Mean Contact Duration"),
        ("Lipid_Count", "bs_lipidcount", "Lipid Count",
         "Binding Site Mean Lipid Count"),
        ("Residence_Time", "bs_restime", f"Residence Time ({ylabel_t})",
         "Binding Site CHOL Residence Time"),
        ("Stability", "bs_stability", "Stability Score (0-1)",
         "Binding Site Pocket Stability Score"),
    ]
    for name, key, ylabel, title in bs_cfg:
        fig, ax = plt.subplots(figsize=(8, 4.5))
        for run_idx, (run_name, res) in enumerate({"Run1": r1, "Run2": r2}.items()):
            vals = [v if not (isinstance(v, float) and np.isnan(v))
                    else 0 for v in res[key]]
            bs_labels = [f"BS{i}" for i in res["bs_ids"]]
            xpos = np.arange(len(bs_labels))
            offset = (run_idx - 0.5) * 0.35
            ax.bar(xpos + offset, vals, 0.35,
                   label=run_name,
                   color=RUN_COLORS[run_name],
                   edgecolor="white", linewidth=0.4,
                   hatch=RUN_HATCHES[run_name])
            for xi, (lbl, val) in enumerate(zip(bs_labels, vals)):
                ax.annotate(lbl, (xi + offset, val),
                           textcoords="offset points",
                           xytext=(0, 5), fontsize=8, ha="center")
            ax.set_ylabel(ylabel, fontsize=11)
            ax.set_title(f"{title}\nRun 1 vs Run 2 | {subunit_name}", fontsize=11)
            ax.legend(fontsize=10)
            ax.grid(axis="y", alpha=0.3)
            plt.tight_layout()
            plt.savefig(fcomp(f"Comparison_BS_{name}.{fig_format}"), dpi=200)
            plt.close()
            logger.info(f"  Saved: Comparison_BS_{name}.{fig_format}")

def plot_summary_heatmap(sel_resids, r1, r2, fname):
    metrics = [
        (f"Run1  Occupancy (%)", r1["res_occupancy"]),
        (f"Run2  Occupancy (%)", r2["res_occupancy"]),
        (f"Run1  Duration ({ylabel_t})", r1["res_duration"]),
        (f"Run2  Duration ({ylabel_t})", r2["res_duration"]),
        (f"Run1  Lipid Count", r1["res_lipidcount"]),
        (f"Run2  Lipid Count", r2["res_lipidcount"]),
        (f"Run1  Residence Time", np.nan_to_num(r1["res_restime"], nan=0)),
        (f"Run2  Residence Time", np.nan_to_num(r2["res_restime"], nan=0)),
        (f"Run1  Stability Score", r1["stability_score"]),
        (f"Run2  Stability Score", r2["stability_score"]),
    ]
    data = np.array([m[1] for m in metrics])
    data_norm = np.zeros_like(data)
    for i, row in enumerate(data):
        rng = row.max() - row.min()
        data_norm[i] = (row - row.min()) / (rng + 1e-12)

```

```

fw = max(8, len(sel_resids) * 1.2)
fig, ax = plt.subplots(figsize=(fw, 5.5))
im = ax.imshow(data_norm, aspect="auto", cmap="YlOrRd", vmin=0, vmax=1)
ax.set_xticks(np.arange(len(sel_resids)))
ax.set_xticklabels([res_label(r) for r in sel_resids],
                    rotation=45, ha="right", fontsize=10)
ax.set_yticks(np.arange(len(metrics)))
ax.set_yticklabels([m[0] for m in metrics], fontsize=9)
ax.set_title(
    f"All Metrics Summary – Run 1 vs Run 2\n"
    f"{subunit_name} | Normalised per row | Darker = higher value",
    fontsize=11)
fig.colorbar(im, ax=ax, label="Normalised value (0-1)", shrink=0.6)
plt.tight_layout()
plt.savefig(fname, dpi=200, bbox_inches="tight")
plt.close()
logger.info(f"    Saved: {fname}")

# =====
# RUN ANALYSES
# =====

script_start = time.time()
logger.info(f"CHOL pocket residue analysis | {subunit_name} | Run 1 vs Run 2")
logger.info(f"Output root      : {output_root}/")
logger.info(f"Log file           : {log_filename}")
logger.info(f"Residues            : {[res_label(r) for r in residues_of_interest]}")
logger.info(f"Cutoffs             : {cutoff_low_ang:.2f} / {cutoff_high_ang:.2f} Å")
logger.info(f"dt                  : {dt_traj:.0f} ps → {dt_traj * time_factor:.6f} {timeunit}/frame")
logger.info(f"Stability           : {n_stability_blocks} blocks")

log_step(1, f"Analysing Run 1 – {subunit_name}")
r1 = run_analysis("Run1", runs["Run1"]["traj"], runs["Run1"]["top"])

log_step(2, f"Analysing Run 2 – {subunit_name}")
r2 = run_analysis("Run2", runs["Run2"]["traj"], runs["Run2"]["top"])

sel_resids = r1["sel_resids"]

# — Comparison plots —————

log_step(3, "Side-by-side bar charts – all metrics")
metric_comp = [
    ("res_occupancy", "Occupancy (%)", "CHOL Occupancy per Residue"),
    ("res_duration", f"Duration ({ylabel_t})", "Mean CHOL Contact Duration per Residue"),
    ("res_lipidcount", "Lipid Count", "Mean Simultaneous CHOL Contacts per Residue"),
    ("res_restime", f"Residence Time ({ylabel_t})", "CHOL Residence Time per Residue"),
    ("stability_score", "Stability Score (0-1)", "Pocket Stability Score per Residue"),
]
for key, ylabel, title in metric_comp:
    v1 = np.nan_to_num(r1[key], nan=0)
    v2 = np.nan_to_num(r2[key], nan=0)
    slug = key.replace("res_", "").replace("_score", "").title().replace(" ", "_")
    plot_comparison_bar(
        sel_resids, v1, v2, ylabel, title,
        fcomp(f"Comparison_{slug}_bar.{fig_format}")

log_step(4, "Difference plots – Run2 minus Run1")
for key, ylabel, title in metric_comp:
    v1 = np.nan_to_num(r1[key], nan=0)
    v2 = np.nan_to_num(r2[key], nan=0)
    slug = key.replace("res_", "").replace("_score", "").title().replace(" ", "_")

```

```

    plot_comparison_diff(
        sel_resids, v1, v2, ylabel, title,
        fcomp(f"Comparison_{slug}_diff.{fig_format}"))

log_step(5, "Distance over time overlay")
plot_comparison_distance(
    sel_resids,
    r1["dist_matrix"], r2["dist_matrix"],
    r1["time_axis"], r2["time_axis"],
    r1["res_occupancy"], r2["res_occupancy"],
    fcomp(f"Comparison_distance_over_time.{fig_format}"))

log_step(6, "Pocket stability comparison")
plot_stability_comparison(
    sel_resids, r1, r2,
    fcomp(f"Comparison_Stability_Score_bar.{fig_format}"),
    fcomp(f"Comparison_Stability_blocks.{fig_format}"),
    fcomp(f"Comparison_Stability_heatmap.{fig_format}"))

log_step(7, "Binding site comparison")
if len(r1["bs_ids"]) > 0 or len(r2["bs_ids"]) > 0:
    plot_bs_comparison(r1, r2)
else:
    logger.warning("No binding sites in either run - skipping BS comparison")

log_step(8, "Summary heatmap")
plot_summary_heatmap(
    sel_resids, r1, r2,
    fcomp(f"Comparison_Summary_Heatmap.{fig_format}"))

log_step(9, "Saving combined CSV")
df_combined = pd.DataFrame({
    "Residue": [res_label(r) for r in sel_resids],
    "Residue ID": sel_resids,
    "R1 Occupancy (%)": np.round(r1["res_occupancy"], 3),
    "R2 Occupancy (%)": np.round(r2["res_occupancy"], 3),
    f"R1 Duration ({timeunit})": np.round(r1["res_duration"], 8),
    f"R2 Duration ({timeunit})": np.round(r2["res_duration"], 8),
    "R1 Lipid Count": np.round(r1["res_lipidcount"], 4),
    "R2 Lipid Count": np.round(r2["res_lipidcount"], 4),
    f"R1 Residence Time ({timeunit})": np.round(r1["res_restime"], 6),
    f"R2 Residence Time ({timeunit})": np.round(r2["res_restime"], 6),
    "R1 Stability Score": np.round(r1["stability_score"], 4),
    "R2 Stability Score": np.round(r2["stability_score"], 4),
    "Δ Occupancy (R2-R1)": np.round(r2["res_occupancy"] -
r1["res_occupancy"], 3),
    f"Δ Duration (R2-R1)": np.round(r2["res_duration"] -
r1["res_duration"], 8),
    "Δ Lipid Count (R2-R1)": np.round(r2["res_lipidcount"] -
r1["res_lipidcount"], 4),
    "Δ Stability (R2-R1)": np.round(r2["stability_score"] -
r1["stability_score"], 4),
})
df_combined.to_csv(fcomp("Comparison_AllMetrics.csv"), index=False)
logger.info(f" Saved: Comparison_AllMetrics.csv")
logger.info(f"\n{df_combined.to_string(index=False)}")

# — Final summary —————
logger.info("")
logger.info("=" * 60)
logger.info(f" ALL DONE | {subunit_name} | total: {elapsed(script_start)}")
logger.info(f" Output: {output_root}/")
logger.info(f" Log : {log_filename}")
logger.info("=" * 60)

```