Kv3 channels in the murine lumbo-sacral spinal cord

Pierce Nathan Mullen

Submitted in accordance with the requirements of the degree of Doctor of Philosophy

The University of Leeds

School of Biomedical Sciences

September 2018

The candidate confirms that the work submitted is on his own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgements

First and foremost, I would like to thank my supervisors, Sue, Jim, Nadia and Charles. Thank you for your mentorship and support, and for putting up with me for four years! I have learned a lot from each of you and thank you for this opportunity.

Thanks to all the Deuchars lot, past and present, Lauryn, Jess, Aaron, Kaisan, Claudia, Varinder, Beatrice, Christian, Cat, Yusoff, Nazlah, Nurha, Norah, Emily, Hanan, Brenda, Lucy and Aun, for making this a fun and lively place to work.

To Jamie and Jon, thanks for some fruitful ion channel discussions and to Manuela e Michele, grazie mille per l'esperienza Italiana!

Thanks to Mum and Dad, I owe them everything and am where I am today because of them. To family, Dan, Caitlin and Hannah, and to friends, Simon, Cat, Kat, Evie, Ed, Cait, Haylee and Liam, you've all shaped me as a person and thanks for making life outside of work fun especially when things weren't going well inside.

And finally to my partner in crime, Sian, for putting up with the long work days and weeks, and for the months I've spent away. You motivate me to be better and I love the life we have created for ourselves.

Abstract

Ion channels are important in a range of physiological processes and can be targeted pharmacologically and therapeutically. Kv3 channels are voltage-gated potassium ion channels important in neuronal firing and synaptic transmission and are highly expressed in the brain and spinal cord. The main aim of this thesis was to investigate the role of Kv3 channels in the spinal cord and we did this in three ways.

Using fluorescence immunohistochemistry we identified, for the first time, expression of Kv3 subunits in the murine lumbosacral spinal cord, at the level of neuronal circuitry that regulates bladder function. Specifically, some of this expression could be attributed to both excitatory and inhibitory synaptic structures closely apposed to bladder motoneurones, the final output neurones in the control of bladder function. Kv3 expression at these locations was susceptible to ageing and was reduced in aged mice. Kv3 channels were functional in synapses as Kv3 blockade with TEA increased the amplitude of the post-synaptic response.

To determine the role of Kv3 channels in a function of the spinal cord, specifically, control over bladder function, we used a modulator AUT1 (Autifony Therapeutics Ltd), which is selective for Kv3 channels, Treatment with AUT1 reduced bladder output in a dose-dependent manner, acutely in young mice and chronically in aged mice suggesting involvement of Kv3 channels in bladder output.

The effect of AUT1 on specific Kv3 subunits was determined in HEK expression cell lines where it was found to modulate both a previously unexplored subunit (Kv3.4a) and a physiologically relevant heteromer. In lumbosacral spinal cord slices, AUT1 suppressed the excitability of interneurones, suggesting that the reduction in bladder output could be occurring at the level of interneurones in the lumbosacral spinal cord.

Modulating Kv3 channels in this way may be a viable therapeutic strategy for conditions presenting with an overactive bladder.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ChAT	Choline acetyltransferase
CNS	Central nervous system
DLN	Dorsolateral nucleus
DRG	Dorsal root ganglion
EUS	External urethral sphincter
EPSP	Excitatory postsynaptic potential
FG	Fluorogold
IML	Intermediolateral column
i.p	Intra-peritoneal
s.c	Subcutaneous
IF	Immunofluorescence
M	Molar
PFA	Paraformalehyde
PMC	Pontine micturition centre
VGluT1	Vesicular glutamate transporter 1
VGluT2	Vesicular glutamate transporter 1
VGAT	Vesicular GABA transporter
GlyT2	Glycine Transporter 2
µm	Micron
Kv	Voltage-gated potassium ion channel
AP	Action potential
AHP	Afterhyperpolarisation
ISI	Interspike Interval
CV	Coefficient of variation

Chapter 1

1 General Introduction

1.1 Ion channels

To understand ion channels it is important to define what ions and ion channels are and how they arise. Ions are salts that have been dissolved in a solvent such as water. This process is known as hydration and arises due to the unequal distribution of electrons and their negative charge within an individual water molecule, which is essentially a single oxygen atom bonded to two hydrogen atoms (H_20). The unequal distribution of negative charge creates regions of a water molecule that spontaneously either have a partial positive or partial negative charge known as dipoles, and similar to the poles of a magnet, opposites attract and equals repel. A salt is typically formed when an atom with an "extra" electron in its outer shell e.g. Na, donates this electron to the shell of an atom with "room" for one more electron in its outer shell, e.g Cl. This leaves a net positive charge at the centre of the Na atom and creates a net negative charge at the Cl atom and an overall stable electron configuration known as an ionic bond. This means that, when a salt and water are mixed together, the negative and positive dipoles of the water molecule are so strongly attracted to the negative and positive charges of a NaCl salt, that the ionic bond breaks and each atom dissociates into solution as free particles that retain their net charge e.g Na^+ and Cl^- ensheathed by a shell of water molecules **(Fig. 1.1)**. These free particles are called ions (Hille, 2001).

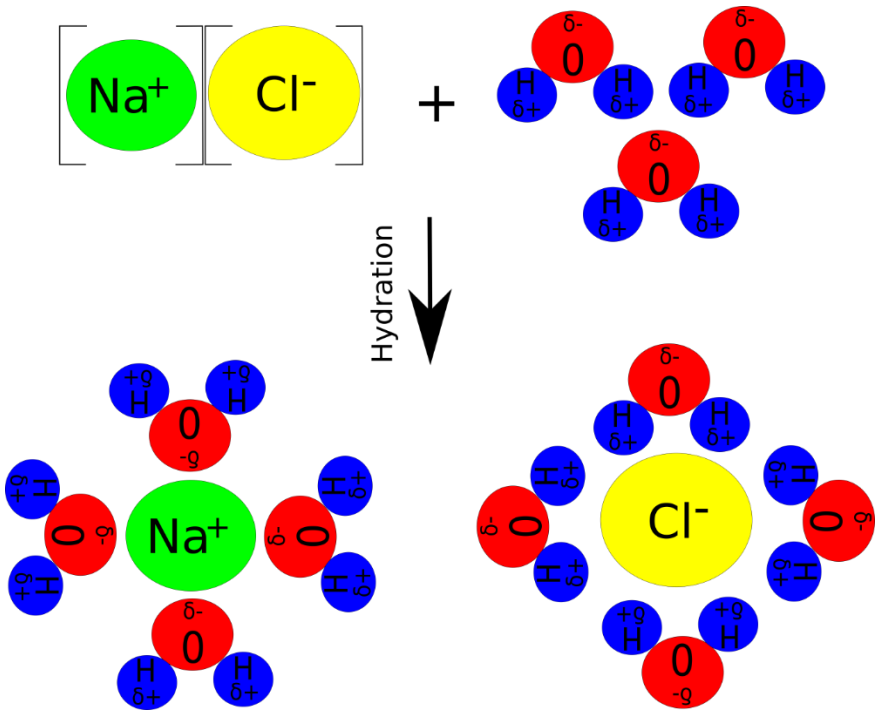


Figure 1.1. Hydration of NaCl.

Dissolving a salt in water produces charged atoms called ions ensheathed in a hydration shell of water molecules. NaCl dissolved in water produces sodium and chloride ions with the negative dipole of water attracted to sodium ion and the positive dipole attracted to chloride.

There are tens of trillions of cells in the human body and each one sits in a relatively aqueous environment and is essentially a lipid envelope containing the machinery of life, water and of relevance to this thesis, free ions dissociated in that water. These ions are numerous and are vital for many processes in the body. Of interest to us here is the complex phenomena that arise when ions such as $Ca2^{+}$, Cl^{-}, K^{+} and Na^{+}, move across the cell from inside (intracellular) to outside (extracellular) and vice versa. As discussed, ions require hydration by water to remain in solution and are hydrophilic, however, the boundary between the intracellular and extracellular of a cell is primarily composed of lipids which exclude water and are hydrophobic. To overcome this and allow the passage of ions, evolution has provided several solutions. Ultimately, by

embedding proteins that span the lipid membrane and cross into both the intracellular and extracellular, nature has created molecular pumps, transporters, exchangers and importantly pore-forming channels, that all work or have work done on them to move ions from one side to the other (Hille 2001). These pore forming channels are ion channels and they endow lipid membranes with permeability; a porous capacity that allows movement of ions and thus a flow of charge from one side to the other. Most ion channels are relatively selective, preferring certain ions over others with a direction of movement established by electrical and chemical gradients, often tailored towards the function required of the membrane in which it sits. Herein lies the origin of the complex phenomena that arises when there is a difference in concentration of an ion between the intracellular and extracellular and also a channel that allows passage of that ion under certain conditions i.e state-dependent permeability.

1.2 Potassium channels

An example is the potassium (K^+) ion channel. These channels are able to fulfil a variety of roles, a result of their great capacity for diversity endowed by the ability to form heteromultimers, to complex with ancillary subunits and to be alternatively spliced. Since the ground-breaking work of Hodgkin and Huxley (1952) in elucidating the sodium and potassium current in the squid giant axon, potassium channels have been implicated in pace-making, cell volume regulation, secretion and proliferation. In the CNS and the neuronal membrane they are vital for the maintenance of the electrochemical gradient from which the state of excitability of a neuron is derived (Pongs 1999).This diversity in

function is achieved by the variability in the properties of the ion channel; the voltage dependence, gating properties and context of its expression ultimately determining the channels influence on the membrane potential. This diversity is highlighted in the large number of families and classes into which the potassium channel can functionally and structurally be divided; sodium (Na+) activated, inward rectifying, 'leak', calcium (Ca2+) activated and voltage-gated, with this thesis focussing on a sub-group within the latter (Coetzee et al. 1999). The voltage-gated K+ class was first cloned in Drosophila melanogaster and can be further divided into families shaker-related (Kv), ether a go go (eag)-related and KvLQT1-related. The shaker-related family was further subdivided into shaker, shab, shaw and shal each respectively homologous to the mammalian K+ channels Kv1.1-1.7, Kv2.1-2.2, Kv3.1-3.4 and Kv4.1-4.3 (Jan and Jan 1997).

1.3 Kv3 channels

Of particular interest for this thesis is the Kv3 family, a family unique in their biophysical properties and ability to endow neurons with high frequency and accurate firing. Their rapid activation and deactivation kinetics and relatively positive activation voltage confer an ability to rapidly repolarise the neuronal membrane during depolarisation, facilitating a faster sodium channel recovery, rapid after-hyperpolarisation phase and fast firing (well reviewed in Rudy et al, 1999, Rudy et al, 2001, Kaczmarek and Zhang, 2017).

1.3.1 Transcription and translation of Kv3 channels

An ion channel is a protein that is the product of the transcription, translation and post-translational modification of a genetic code that resides within a cell nucleus **(Fig. 1.2)**. The gene encoding each ion channel has a chromosomal

position or genetic locus from which this process of ion channel synthesis arises. In humans, the loci of the Kv3 subunits have been identified with Kv3.1 (KCNC1) at 11p15, Kv3.2(KCNC2) at 12q14.1, Kv3.3(KCNC3) at 19q13.33 and Kv3.4 at 1p21(KCNC4) and for mice at chromosome 7, 10, 7, 7/10, respectively (Coetzee et al. 1999) . From these loci the transcription of KCNC genes occurs. Transcription is controlled by regulatory sequences and the promoter of the gene. A negative element in the 5` untranslated region of the promoter for KCNC1 (Kv3.1) in addition to an enhancer in the promoter have been described to determine expression (Gan et al, 1999). Transcription is also regulated by cAMP-response element binding protein (CREB) with promoter activity favoured by increased intracellular concentrations of cAMP and Ca^{2+} that occur during neuronal activation. Transcription of KCNC (Kv3) genes produces mRNA and this mRNA can be bound and subsequent translation inhibited by the Fragile X mental retardation protein (FMRP) (Strombos et al, 2010). In this study inhibition was reversed by increased neuronal activity. Together, the transcription of KCNC genes and translation of Kv3 mRNA is tightly coupled to the intensity of neuronal activity, facilitating the ability of the neurone to respond to high levels and rates of stimulation.

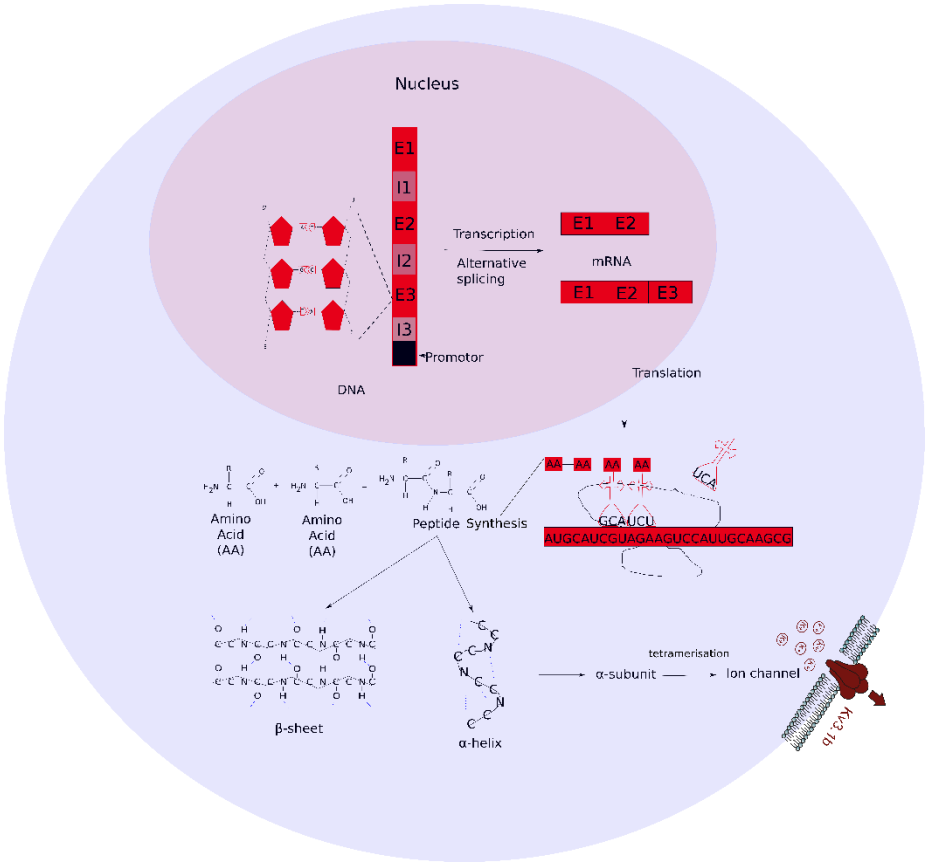


Figure 1.2. A general overview of Kv3 channel synthesis.

Alternative splicing occurs during transcription producing different variants of the same gene. Translation at ribosomes synthesises peptides that can be arranged in β sheets and α helices to form a subunit with specific domains. Subunits tetramerise to form a complete membrane channel.

1.3.2 Kv3 structural identity: the consensus and the diversity endowed by alternative splicing, β subunits, heteromers and the mechanisms of gating.

Our knowledge of the structure of the Kv3 ion channel is largely inferred from the crystal structures of the bacterial KcsA channel (Doyle et al. 1998), the archaea KvAP channel (Jiang et al. 2003) and the mammalian Kv1.2 channel (Long, Campbell and Mackinnon 2005). Whilst variability exists between the conformational states (e.g. open, closed or bound by ancillary subunit) and the integrity of the membranes in which these crystal structures sit, a consensus on general structure has been reached. These studies derive a tetrameric structure composed of four Kv3α-subunits, each subunit consisting of 6 transmembrane segments denoted S1-6 **(Fig. 1.3)**. This structure is divided into two domains, a voltage sensing domain S1-S4 and a pore forming domain (S5-S6) attached to the former by a S4-S5 linker. The structure consists of two gates, an upper formed by a P loop between S5 and S6 and a lower formed by the S6 helix. The selectivity of potassium channels arises from a highly conserved five residue electronegative sequence, TVGYG, within the P-loop that promotes dehydration of potassium ions and entry into the pore. Each Kvα subunit represented by Kv3.1-Kv3.4 can assemble with β subunits, such as those in encoded by the KCNE genes (ancillary subunits Mink, MiRP1and MiRP2) to permit modulation. The channel's inactivation is either of the slow C-type mediated via the upper gate or fast N-type mediated by the N-terminus (Grizel, Glukhov and Sokolova 2014). Each subunit also contains an intracellular C-terminus that varies between subunits and splice variants of these subunits. This alternative splicing occurs at the 3' end to produce Kv3.1a-b, Kv3.2a-d, Kv3.3a-c and Kv3.4a-c (Luneau et al. 1991). The Kv3 structure also contains an intracellular N-

terminus with a T1 domain that permits tetramerisation and assembly of four Kvα subunits to form the channel (Xu et al. 1995; Choe 2002). This tetrameric structure can exist either in a homomeric or a heteromeric formation, the latter occurring only between Kvα subunits within the same subfamily (Rettig et al. 1992). Ultimately this tetrameric structure facilitates a channel that is K+ selective, voltage sensitive and diverse in its biophysical properties and expression as a result of alternative splicing and assembly.

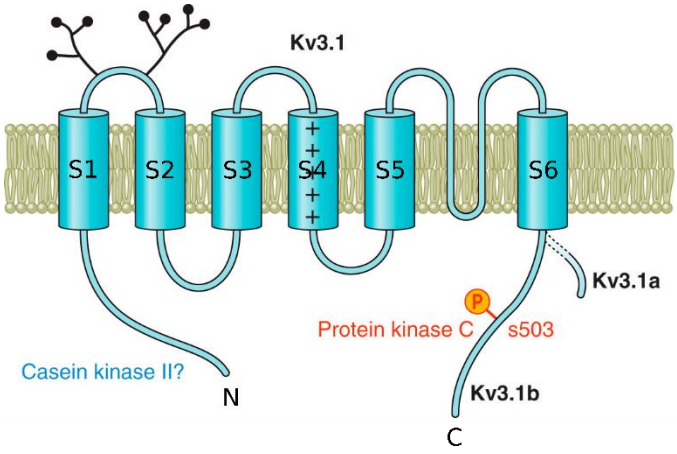


Figure 1.3. Kv3α subunit (adapted from Kaczmarek and Zhang, 2017).

A Kv3.1 subunit shows six transmembrane domains; S1-4 represents the voltage sensing domain and the S5-S6 represent the pore domain. S4 contains charges that move during changes in the electric field, changing the conformation of the channel from open to closed or vice versa. Phosphorylated residues are also depicted.

1.3.3 The Gating of Kv3 Channels

In understanding the Kv3α structure we must also understand how it permits the coupling of changes in the electrical field with the mechanical work required to allow flux of K+ ions from the intracellular to the extracellular; in other words how the voltage defines the open, closed or intermediate conformational state of the channel. The underlying principle behind this coupling is the

movement of charged amino acids within the structure (e.g. the movement of the positively charged S4 segment) as a result of force exerted by the electrical field. This is where consensus largely dissolves. Despite the weak electron density in segments that underlie the voltage sensing region, lots of models have attempted to explain the mechanism of the voltage sensor with each postulating different degrees and planes of S4 movement (for extensive review see Grizel, Glukhov and Sokolova (2014)). Controversy, in particular, exists between the translational helical model and the paddle model of S4 movement. The translational model proposes, based on eukaryotic studies, that the electric field is focussed on a small part of the S4 segment due to the aqueous vestibules that surround the segment, forming a gating pore, and that this minimises the energetic cost and distance of translocation of S4 charges within the lipid membrane (Catterall, 2010). The paddle model, however, argues, based on prokaryotic studies (Mackinnon, 2003) that the S4 segment and S3b segments translocate across the whole thickness of the membrane. The problem with the paddle model is that its conception is based on the prokaryotic KvAP channel which displayed the S4 segment resting on its side near the intracellular interior. This combined with binding studies that found interaction with the S4 segment during activation on the extracellular side suggested a paddle-like movement across the membrane to expose S4 during activation. However, the crux of the issue lies in the lack of a closed-state mammalian Kv structure; only the open structure of Kv1.2 and a chimaera have been detailed (Long et al, 2005, 2007). The issue lies in whether the S4 segment is in a transmembrane position during both activation and rest; if it is transmembrane then the paddle model doesn't fit however a closed-state structure, ultimately, is required to

confirm this. Functional evidence contradicts the paddle model. An array of toxin binding studies find that the N-terminal end and S3-S4 linker are accessible from the extracellular side in the resting and activated state consistent with transmembrane orientation in both states (Catterall, 2010). Because of the lack of a closed-state structure, all-atom molecular dynamic simulations have been used to model the channel gating from the open Kv1.2 /2.1 chimaera channel to a closed state (Jensen et al, 2012). These simulations support the idea of movement through a gating pore as described by the sliding-helix model. Interestingly, introducing mutations in S4 residues in these simulations produced a leak current consistent with the idea of an aqueous gating pore. In addition, the paddle model doesn't appear to account for the energetic cost of such a translocation, however, the sliding helix model proposes the sequential formation of ion pairs between S4 positive charges and other residues on transmembrane segments as a less costly route through the membrane (Catterall, 2010). In summary, the paddle model probably describes voltage sensing in the KvAP channel or crystallised form used for structural studies, however, for mammalian Kv channels much of the functional evidence supports the sliding helix model (or one similar to it) of voltage-dependent gating.

1.3.4 The Biophysical Properties of Kv3 Channels

An action potential is produced as a result of numerous ionic conductances across the membrane mediated by a variety of ion channels. These channels facilitate the depolarisation, repolarisation, hyperpolarisation and afterhyperpolarisation of the neuronal membrane. Kv3 channels are widely

regarded as being high-voltage-activated having a depolarised activation voltage and rapid activation and deactivation kinetics. This means that Kv3 channels are activated during membrane changes seen during action potentials to actively repolarise the membrane **(see Fig. 1.4)** However, in identifying the biophysical properties of the Kv3 channels one encounters some disparities, mainly arising from results in heterologous systems and those in native cells, with the debate centred on whether the former reflects the complexity and reality of the latter. In a review by Rudy et al. (1999) arguing that currents measured in expression systems resemble those in native Kv3 channels, it is stated that Kv3 channels are activated at -10mV. This appears to be later amended to -20mV in a review by the same authors in the same expression systems (Rudy and McBain 2001). Conversely, Baranauskas et al. (2003) cite -30mV as the activation voltage whilst proposing heteromerisation as a solution to the disparity. Using in situ hybridisation, co-immunoprecipitation and patch clamp they showed that Kv3.1 and Kv3.4a are co-expressed and co-assembled in rat fast spiking interneurones in the globus pallidus, hippocampus and subthalamic nucleus to shorten the spike duration and enhance spike rate. However in some fast spiking neurons such as within the medial nucleus of the trapezoid body (MNTB) Kv3.4 is notably absent suggesting that Kv3.4 is not required for fast firing where Kv3.1 expression suffices (Wang et al, 1998). Interestingly, Martina, Yao and Bean (2003) observed an activation voltage of -10 mV for putative Kv3.3 and Kv3.4 subunits in rat cerebellar purkinje cells using outside out patches. According to Rudy et al. (1999) in HEK cells, Kv3.1, Kv3.2 and Kv3.3 possess non-inactivating delayed rectifier properties whereas Kv3.4 exhibits a transient A-type current, however, the Kv3.3 current is

transient in oocytes (Rettig et al. 1992) and in Rudy and McBain (2001) they noted that in CHO cells the Kv3.3 current is a transient A-type current with fast inactivation. The reason for this difference in inactivation for Kv3.3 between HEK cells and CHO cells has been postulated to be due to incorrect translation in HEK cells (Fernandez et al, 2003). The rate of deactivation at -60 mV has been reported as less than 2 ms and the rate of activation at +40 mV as less than 20 ms (Coetzee et al, 1999). Evidently, the activation voltage and rate of inactivation differs depending on the expression system used and likely varies in native cells in the CNS, depending on the context of expression and intrinsic modulation.

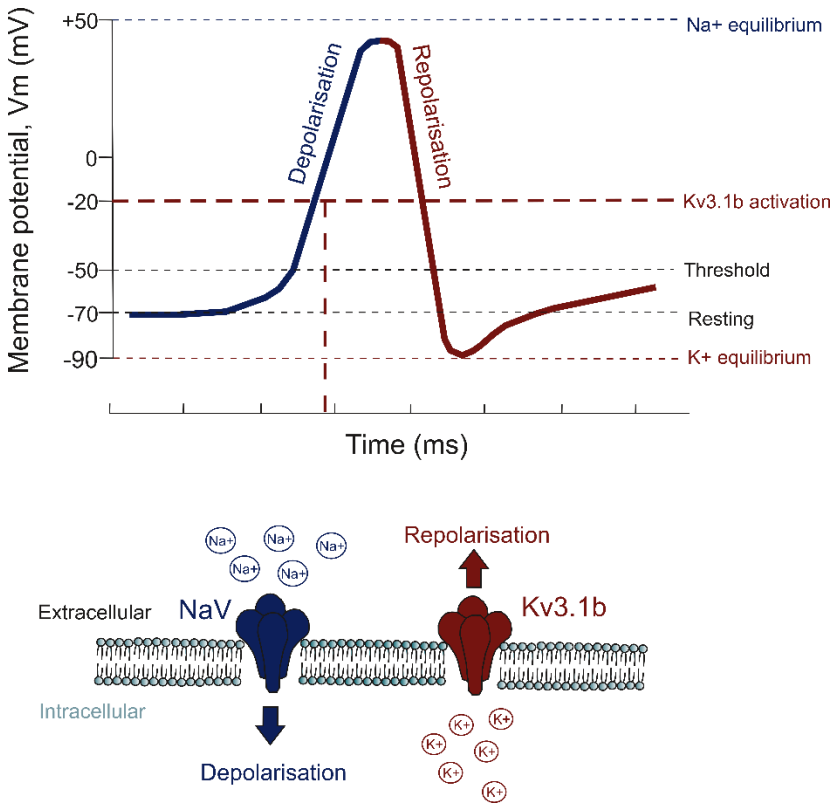


Figure 1.4. Kv3 channel activation.

Kv3 channels are activated at potentials, ~ -10 mV, achieved during action potentials with a peak conductance during the repolarisation phase. During repolarisation potassium ions flow out of the cell.

1.3.5 Expression mRNA to Protein, Embryo to Adult

Kv3 channels are widely distributed and display an overlapping pattern throughout the rodent CNS **(see Fig. 1.5).** Using Northern blotting and in situ hybridisation in adult rats, Weiser et al. (1994) found Kv3.1, Kv3.2 and Kv3.3 transcripts within the CNS. Kv3.2 transcripts were present in the dorsal thalamus, while the Kv3.1 and Kv3.3 signal was strongest in the spinal cord, cerebellar cortex, inferior colliculus and olfactory bulb. Kv3.4 transcripts were mainly in skeletal muscle whilst low levels of the transcript in the CNS in a pattern often overlapping with Kv3.1 and Kv3.3 suggests the possibility of heteromer formation. Rettig et al. (1992) further elaborated using Northern

blot in postnatal day 60 (p60) rats identifying Kv3.4 in the dentate gyrus and a granulated pattern of Kv3.1 and Kv3.2 in the hippocampus. Weiser et al. (1995) furthered these expression studies using Kv3.1 antibodies. Kv3.1 immuno labelling largely agreed with the previous mRNA distribution studies by Weiser and Rettig with some discrepancies occurring in the olfactory bulb (no protein present) and the cerebellum (protein in the molecular layer as well as the granule layer). Conversely, Ozaita et al. (2002) did find presence of Kv3.1a protein within the olfactory bulb. Subcellularly, Kv3.1b was localised to the somatic, proximal dendritic and axonal membranes in interneuron-like parvalbumin-positive cells in the hippocampus, with this parvalbumin colocalisation also seen by McDonald and Mascagni (2006) in the basolateral amygdala. Chang et al. (2007) reported an expression pattern of Kv3.3 protein in mice, again consistent with Weiser et al. (1994), in the auditory brainstem and purkinje and granule cells of the cerebellar cortex usually within parvalbumin-positive and inhibitory interneurones. Kv3.3 exhibited a subcellular distribution throughout the neuron, in the membranes of distal dendrites, axons, somata and terminals. They also observed overlapping patterns between Kv3.3 and Kv3.1 and also between Kv3.3 and Kv3.4 further suggesting the existence of native heteromers. At the level of the brainstem, the expression patterns of Kv3 have been defined. Kv3.3 is predominately expressed in most auditory neurons whilst Kv3.1 is expressed in the calyx of Held and tonotopically in the MNTB (Li, Kaczmarek and Perney 2001; Elezgarai et al. 2003). Kv3.1 is also expressed in the gracile, cuneate and spinal trigeminal nuclei, and presynaptically in the nucleus tractus solitarius (NTS) (Dallas et al. 2005). Kv3.3 co-localises with Kv3.1 within these nuclei in terminals also

double labelled for vesicular glutamate transporter (VGluT2) and the glycine transporter (GlyT2) suggesting presence of Kv3 channels in both excitatory and inhibitory neurons (Brooke et al. 2006). Kv3.3 is expressed in medial vestibular nuclei in somatic, dendritic and terminal membranes again co-localising with vGluT2 and GlyT2. Electron microscopy elucidated a postsynaptic expression and vGluT1 positive terminals onto and enclosing Kv3.3-positive soma (Brooke et al. 2010). Kv3.4 exhibits immunoreactivity pre and post-synaptically in the dorsal vagal nucleus (DVN), nucleus ambiguus (NA) and NTS (Dallas et al. 2008; Brooke et al. 2004a).

In the spinal cord at the lumbar level Kv3.1b is expressed in Renshaw cells in laminae VII (Song et al. 2006), in GAD65 and GAD67-positive neurons in laminae I-III of the dorsal horn (Nowak et al. 2011) and in interneurones on the periphery of the intermediolateral nucleus (IML) that are antecedent to Kv3-negative sympathetic preganglionic neurons (SPN) (Brooke et al. 2002; Deuchars et al. 2001). Interestingly, at the subcellular level, Kv3.1b is observed in the nodes of large myelinated axons in the rat and murine spinal cord (Devaux et al. 2003). Kv3.3 at the thoracic level is expressed presynaptically in lamina IV, V, in the dorsal and ventral horn and the central canal, and co-localises with Kv3.1b in these areas. In the ventral horn, electron microscopy showed that this Kv3.3 immunoreactivity was not in the somata of motoneurones (Brooke et al. 2006). Kv3.4 exhibits immunoreactivity pre and post-synaptically in the IML and ventral horn. Kv3.3 and Kv3.4 protein is also present in the prejunctional terminal of the neuro-muscular junction (NMJ) (Brooke et al. 2004b). Kv3 channels are also expressed in non-excitable cells;

Kv3.1 underlies the I current in T lymphocytes from lymph nodes of auto immune disease mouse models and a human lymphoma cell line (Grissmer et al. 1992).

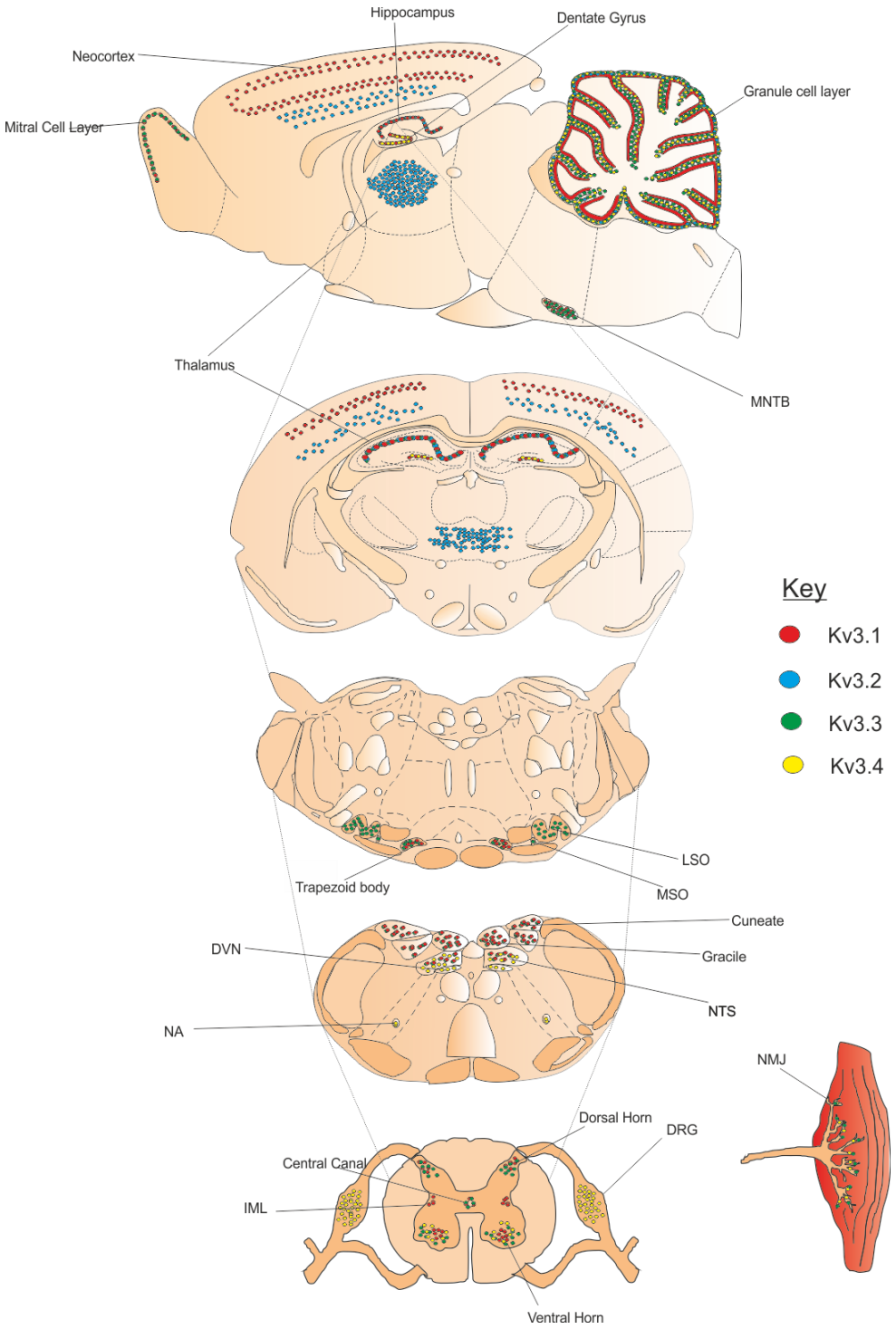

Hippocampus
Dentate Gyrus
Neocortex
Granule cell layer
Mitral Cell Layer
Thalamus
MNTB
Key
Kv3.1
Kv3.2
Kv3.3
Kv3.4
LSO
Trapezoid body
MSO
Cuneate
DVN
Gracile
NTS
NA
NMJ
Dorsal Horn
DRG
Central Canal
IML
Ventral Horn

Figure 1.5. A general representation of select expression patterns for Kv3 subunits.

At the level of the brain, brainstem and spinal cord, Kv3 subunits, Kv3.1 (red), Kv3.2 (blue), Kv3.3 (green), Kv3.4 (yellow), are widely expressed often in an overlapping pattern of expression. Abbreviations; MNTB, medial nucleus of trapezoid body; LSO, lateral superior olive; MSO, medial superior olive; DVN, dorsal vagal nucleus; NA, nucleus ambiguous; NTS, nucleus tractus solitaris; NMJ, neuromuscular junction; IML, intermediolateral nucleus; DRG, dorsal root ganglion.

When assessing the localisation and quantity of a protein, it is important to understand the changes in expression that may be exhibited throughout development; from the embryo to the neonate to the adult. The Kv3.1 channel highlights how expression can change during development and hence asks some important questions as to the regulation of expression of the protein throughout this process. Using in situ hybridisation, Perney et al. (1992) observed predominance of the Kv3.1a splice variant during early development in embryonic neurons but an increase in the Kv3.1b variant from embryonic day 17 (E17) to postnatal day 10 (P10). Kv3.1b transcripts were highest in the neocortex, hippocampus and cerebellum and were co-expressed with Kv3.1a in the same neurons perhaps suggesting intra-splice variant heteromer formation. However, Du et al. (1996) observed differences between the mRNA increase described by Perney et al, (1992) and that of protein expression; the number of Kv3.1b-positive cells increased from P8 to a maximum at P14 whilst the Kv3.1b content in isolated membrane vesicles continued increasing up to P40. Perhaps an increase in translation at later stages of development explains this disparity.

In spite of this, these increases concur on Kv3.1b becoming the predominant splice variant in adults. Kv3.1a and Kv3.1b are biophysically identical so why one preference over another? In cerebellum slices Liu and Kaczmarek (1998) found that depolarisation suppressed up-regulation of Kv3.1a mRNA induced by basic fibroblast growth factor (bFGF) whereas the up-regulation of Kv3.1b remained unaffected. Perhaps an increase in excitable activity and level of depolarisation throughout development results in the predominance of the Kv3.1b variant.

1.3.6 Targeting of Kv3 channels: the role of ancillary subunits, alternative splicing, N-glycosylation and targeting motifs in the subcellular expression of Kv3 channels

Kv3 channels are expressed in an array structures throughout the CNS: localised subcellularly throughout different neuronal types to facilitate the specific function of that neuron. But how is this subcellular expression determined? The targeting and context of an ion channel's expression is crucial in determining the function that channel will play. Kv3 channels are expressed in the membranes of neurons: in dendrites, axons, somata and terminals. The targeting of these channels to these neuronal membranes is dependent on many factors. Xu et al. (2007) identified a lysine rich axon targeting motif (ATM) on the C-termini of Kv3.1a and Kv3.1b channels in cultured hippocampal cells that could interact with the T1 domain on the N-terminus. This interaction was zinc-dependent with mutations at the position H459 in the first extracellular loop and in the zinc binding site in T1 resulting in a reduction in Kv3.1 axonal targeting (Xu et al. 2010; Gu, Barry and Gu 2013). Xu et al. (2007) also identified a role of ankyrin G in the targeting of Kv3 channels to the axonal membrane.

Ankyrin G is an adapter protein and co-localises with Kv3.1b in nodes (Devaux et al. 2003). siRNA knock down and dominant negative mutants of ankyrin G reduced the targeting of Kv3.1a and Kv3.1b to the membranes of axons and dendrites (Xu et al, 2010). This same group also demonstrated a role for the heavy chain of conventional kinesin 1, KIF5, in transporting Kv3.1 through the axon initial segment to the membranes of axons in hippocampal neurone cultures (Xu et al. 2010). Using time lapse microscopy and surface plasma resonance, a direct interaction of nM affinity (Kd=6x10-8M) and 'comovement' of Kv3.1 with KIF5 was observed along the axons of mature hippocampal neurone cultures (Barry et al. 2013).

Ancillary β-subunits, encoded by KCNE genes, can assemble with Kv3 channels and have also been implicated in the regulation of surface expression of these channels. Kv3.4 and Kv3.3 can be retained intracellularly. Kanda et al. (2011a) demonstrated in CHO cells using avidin-biotin purified surface fractions that binding to ancillary β-subunits KCNE1 and KCNE2 prevented surface expression of Kv3.3 and Kv3.4 and resulted in retention in the ER and golgi apparatus. The ancillary subunits also determine the membrane expression of Kv3.1 and Kv3.2. In CHO cells, Kv3.1 and Kv3.2 appeared to prevent KCNE1 coassembly and thus the retention of N-type Kv3 subunits, Kv3.3 and Kv3.4. The group therefore concluded that membrane expression of Kv3.3 and Kv3.4 requires heteromer formation with delayed rectifiers Kv3.1 or Kv3.2 in order to bypass a 'checkpoint' determined by KCNE β-subunits and their assembly with Kvα subunits in the early secretory pathways (Kanda et al. 2011b). This idea suggests that Kv3.3 and Kv3.4 shouldn't exist as homomers however, Kv3.4

subunits exist in isolation of other Kv3 subunits in the dorsal root ganglion (Ritter et al, 2012). This is therefore, an important contrast to be aware of when using overexpression systems as a proxy for native expression.

N-glycosylation has also been implicated in the distribution of Kv3 channels. In B35 neuroblastoma cells normal occupancy of both N-glycosylation sites in the extracellular linker between S1 and S2, produced a distribution throughout the cell outgrowth and body, whereas vacancy of one site resulted in restriction of Kv3.1b to the cell body (Hall et al. 2014)

It has also been suggested that the function of alternative splicing is to regulate the targeting of the channel to the membrane. Kv3.2 isoforms are differentially expressed in polarised MDCK cells; Kv3.2a is targeted to the basolateral side where Kv3.2b and Kv3.2c are localised to the apical membrane (Ponce et al. 1997). Furthermore, Kv3.1 splice variants exhibit differential subcellular expression. Gu et al. (2012) used cultured hippocampal neurons cells to show that the Kv3.1a isoform was restricted to somatodendritic sites whereas the Kv3.1b isoform was predominate in the axonal membrane. They demonstrated that slow spiking neurons did not contain Kv3.1b, whilst transfection using Lipofectamine converted these slow spiking neurons to fast spiking despite the two isoforms being biophysically identical. The increase in spiking frequency was attributable to axonal expression of Kv3.1b. Application of 1mM tetraethylammonium (TEA) to the axonal membrane reduced spiking frequency whereas addition to the proximal dendrite did not. This suggests that whilst Kv3.1b is expressed in the proximal dendrites along with Kv3.1a, its production of the fast spiking phenotype occurs only with and requires axonal expression.

In an effort to understand the variant specific targeting they created deletion constructs of the C-terminus which suggested the presence of an axonal targeting motif masked by the tetramerisation T1 domain of the N-terminus in Kv3.1a but unmasked in Kv3.1b. Deletion did not affect the properties of either channel. Conversely the findings of Ozaita et al. (2002) in hippocampal tissue directly contradict the findings of Gu et al, (2012) suggesting instead that Kv3.1a has an axonal targeting property and is only expressed axonally and presynaptically and targets Kv3.1b to the axonal membrane via heteromerisation. Gu et al. (2012) used transfected hippocampal cultured neurons which would surely contain endogenous low levels of Kv3.1 isoforms. Perhaps endogenous levels of Kv3.1a are sufficient to facilitate the axonal Kv3.1b targeting observed by this group. It would therefore be interesting to see if Kv3.1b exists as a defined homomeric channel at the native axonal membrane. In addition, Ozaita et al. (2002) do note that there are some exceptions to the axonal localisation of Kv3.1a; in the murine mitral cells of the olfactory bulb Kv3.1a is localised to somatodendritic membranes. This perhaps suggests that isoform specific subcellular localisation may vary by cell type.

In brief, several factors appear to affect the subcellular expression of Kv3 channels; alternative splicing may produce different targeting motifs and ankG and KIF5 escort specific Kv3 isoforms along the axon to facilitate axonal fast spiking.

1.3.7 Modulation of Kv3 channels

Modulation of Kv3 channels can occur in a variety of ways and this partially explains the diversity in current and function. Phosphorylation is one of the main forms of modulation of Kv3 channels. Activation of protein kinase C (PKC) removes the N-type inactivation of Kv3.3 in CHO cells and oocytes with mutagenesis studies highlighting sites of phosphorylation at serine positions 3 and 9 (Desai et al. 2008). Song et al. (2005) highlighted a role of phosphorylation in the rat auditory brainstem. High frequency auditory stimuli dephosphorylated the basal phosphorylation of Kv3, the subsequent result being an increase in current attributable to Kv3.1 and the neuron's ability to fire at high frequencies. Accompanied by an ability to fire at higher frequency is a loss in the accuracy of firing, therefore this group postulate that phosphorylation is a mechanism of modulation crucial for the adaptation of Kv3.1 to the level of excitability around it. Using okadaic acid, they identified phosphatases, phosphoprotein phosphatase1 (PP1) and PP2A, as candidates involved in the dephosphorylation of Kv3.1b in MNTB neurons. The question thus arises as to what inputs are required to maintain the basal phosphorylation, likely via PKC, of Kv3.1 in quiescent conditions and as to how high frequency auditory stimulation induces phosphatase pathways such as those of PP1 and PP2A. Cotella et al. (2013) also identified potential phosphatases that could reverse the phosphorylation of Kv3.1. Using c.elegans as a model, they identified a phosphatase with a mammalian homologue, prostatic acid phosphatase a (PAPa). This mammalian homologue, PAPa directly interacted with Kv3.1b in mouse brain. Co-localisation between Kv3.1b and PAPa was interestingly highest in the periventricular progenitor stem cells of

the ventricular region. In CHO cells PAPa reversed (dephosporylated) PKC-mediated phosphorylation of Kv3.1. This perhaps implies that PAPa acts to regulate the phosphorylation of Kv3.1b in certain cell types. The idea that phosphorylation of Kv3.1 allows neurons to adapt their response to different types and levels of stimulation is further supported by the findings of Macica et al. (2003) who found that phosphorylation by PKC at serine 503 improved the timing of firing. Kv3.1 is also basally phosphorylated in CHO cells and in MNTB neurons by casein kinase 2, with phosphorylation in this situation negatively shifting the activation voltage by -20mV (Macica and Kaczmarek 2001). Modulatory negative shifts in activation voltage may explain why Kv3 channels can be active in non-excitable cells such as in T-lymphocytes. Kv3.2 is phosphorylated by protein kinase A (PKA) (Moreno et al. 1995). Phosphorylation of Kv3.2 in fast spiking hippocampal parvalbumin-positive interneurones was mediated by PKA as a result of H2 histamine receptor activation and reduced the firing frequency and that of oscillations in principle cell layers. These effects were absent with Kv3.2 knockout perhaps suggesting a homomeric assembly of Kv3.2 or a heteromer (Kv3.1 and Kv3.2 are present in the some of the same populations) requiring at least one Kv3.2 subunit for its proper function. Kv3.4 is modulated by PKC (Ritter et al. 2012). In this study Kv3.4 was found to be expressed in dorsal root ganglion neurons (DRGs) and unable to inactivate via N-terminal occlusion of the channel pore. This is due to phosphorylation at four serine sites on the N-terminus, S8, S9, S15 and S21, a modulation of inactivation induced by phorbol myristate acetate (PMA) but interestingly also by GPCR agonists (Ritter et al, 2012). The effect of GPCR agonist action was only present when inside the patch electrode which

interestingly appears to suggest the formation of a complex comprising of Kv3.4 and a GPCR. Recently, dysregulation and downregulation of Kv3.4 in DRGs was implicated in pain sensitisation as a result of spinal cord injury (SCI) (Ritter et al, 2015). In this study, spinal cord injury resulted in hyperexcitability of contralateral DRG neurons and a downregulation of the Kv3.4 current. This dysregulation of Kv3.4 was proposed to be responsible for the hyperexcitability of the neuron. The authors presented decreased surface expression, calculating membrane changes of Kv3.4 relative to the intensity of a membrane marker, as the underlying factor behind this downregulation. It is often difficult to ensure accuracy when quantifying expression changes at the membrane in this way, and perhaps this study would have benefited from comparing the basal phosphorylation between naïve and post-SCI animals. In this case it would be interesting to see if an upregulation in Kv3.4 phosphorylation underlies this dysfunctional, slowly inactivating Kv3.4 current and also hyperexcitable state of the DRG neurons.

Kv3 channels can be further modulated at the membrane by lipids and ancillary subunits. Oliver et al. (2004) demonstrated a role for membrane lipids in modulation of Kv3 channels using xenopus oocytes transfected with mammalian channels. Phospoinositides removed Kv3.4 N-type inactivation whilst arachidonic acid and anandamide provided delayed rectifier and slow inactivating Kv3.1 currents with a fast inactivation. The binding of ancillary β-subunits to Kv3α subunits offers another mode of modulation. Mink, MiRP1 and MiRP2 are encoded by the KCNE genes and form complexes with Kv3.1, Kv3.2 and also heteromers of these two channels in CHO cells (Lewis, McCrossan and

Abbott 2004). This binding acted to negatively shift the activation voltage, whilst also slowing the characteristically fast activation and deactivation kinetics of Kv3.1 and Kv3.2 channels. The group suggests that this ability to switch a fast spiking (FS) phenotype and capability of a neuron on and off by modulating Kv3 currents may explain why we find Kv3 in non-fast spiking cells too. It also begs the question as to the role of Kv3 in these non-FS cells, such as in Mink-containing-T-lymphocytes (Grissmer et al. 1992). The interaction (co-immunoprecipitation) between MiRP2 and Kv3.1b has been demonstrated by McCrossan et al. (2003) in native tissue but not in E18 hippocampal neurons. Given that Kv3.1b isn't predominately expressed at this age perhaps it is understandable that a complex wasn't precipitated in this model. Kv3 channels can also be pharmacologically modulated. Kv3 currents are sensitive to 4-aminopyridine (4AP), gambierol, tetraethylammonium (TEA) and quinine and insensitive to dendrotoxin (DTX) to varying degrees (Rettig et al. 1992, Johnston et al 2010).

1.3.8 Kv3 Function: Fast firing, Fidelity and Regulation of Neurotransmitter Release

1.3.8.1 Firing

Within the CNS Kv3 channels are almost exclusively expressed in neurons, with some exceptions such as Kv3.4 expression in astrocytes (Boscia et al, 2017). A neuron functions to generate action potentials, initiating at the axon initial segment and propagating along the axon towards the axonal terminal. The frequency of action potentials is determined by the frequency of supra-threshold stimuli and the ability of a neurone to match that frequency, and this ability is limited by the refractory period and adaptation of the neurone. Action

potentials are propagated along the axon with a waveform dependent on local ion channel expression **(Fig. 1.6)**, terminating at presynaptic terminals to stimulate neurotransmitter release. This section will focus on the various roles that Kv3 plays dependent on its expression, subcellularly within the neuron and cellularly within the CNS.

The first role to discuss is one of facilitating high frequency firing and a fast spiking phenotype. Erisir et al. (1999) observed Kv3.1 and Kv3.2 in mouse fast spiking neocortical interneurones and used a pharmacological approach to assess the role of Kv3 channels in high frequency firing. These interneurones were TEA (1mM) sensitive, application of which resulted in a reduction in firing frequency. The biophysical characteristics of Kv3 channels caused the authors to postulate that the ability of Kv3 channels to rapidly hyperpolarise the membrane upon depolarisation allows for a large fast afterhyperpolarisation and recovery of sodium channels from inactivation, thus conferring the ability to fire at higher frequencies. Lien and Jonas (2003) confirmed this role in FS hippocampal interneurones using computer simulation. By extrapolating the current responsible for the Kv3 channels and by using a fast dynamic clamp system they were able to add or subtract the pharmacologically isolated (4AP and TEA) Kv3 current to hippocampal interneurones and observe the effect on firing. Addition of Kv3 conductance restored the high frequency pattern whereas subtraction removed it.

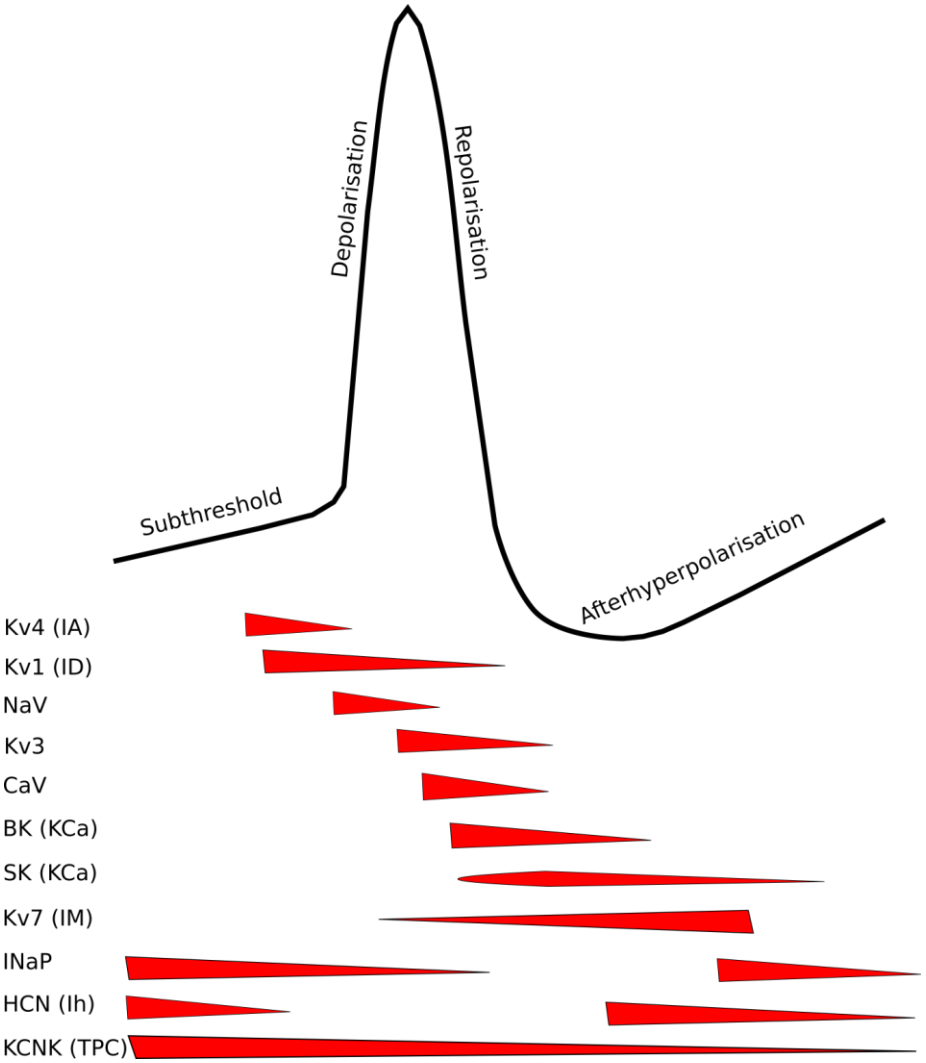


Figure 1.6. A general overview of the conductances, their associated channels and estimated time-course during a schematic action potential.

Kv3 channel conductance peaks during an action potential and deactivates rapidly during the repolarisation phase. The presence of Kv3 channels determines the contribution of other channels by keeping suprathreshold events short and by limiting Ca^{2+} influx. Other potassium and sodium channels that contribute to the action potential waveform are also depicted here.

1.3.8.2 The role of Kv3 channels in synaptic transmission

Kv3 channels also have a role in neurotransmitter regulation, the idea that Kv3 channels act at presynaptic terminals to restrict cytotoxic influx of Ca2+ and synchronise discrete release of neurotransmitter into the synaptic cleft. In retinal ganglion cells, Kuznetsov et al. (2012) demonstrated that TEA application (250 μM-1 mM) increased the Ca2+ amplitude per action potential. The action potential under these pharmacological conditions had an increased width and a reduced frequency. They also showed the presence of Kv3.1 and Kv3.2 mRNA transcripts in these cells, although the expression levels differ from those reported by Ozaita et al. (2002). This regulatory role of Kv3 channels in neurotransmitter release is further supported by Rowan, Tranquil and Christie (2014). Using two photon voltage-sensitive dye imaging and photolysis of caged 4AP to block Kv3 channels at local sites along the neuron, they identified that Kv3 locally determines the repolarisation in presynaptic boutons and that blockade increased Ca2+ influx. Recording from the calyx of Held in brainstem slices, Ishikawa et al. (2003) observed Kv3 currents in this large presynaptic terminal. Application of 1mM TEA prolonged the action potential and increased the peak amplitude. In this setup paired presynaptic and postsynaptic whole cell recordings were able to elucidate a potentiated excitatory postsynaptic current (EPSC) as a result of TEA application. This further suggests a role of Kv3 currents in regulating the release of neurotransmitter from the presynaptic terminal and thus in determining the excitability of postsynaptic structures. Furthermore, Goldberg et al. (2005) demonstrated a TEA-sensitive (1mM) two fold increase in GABA release from neocortical GABAergic fast spiking interneurones. This in combination with a

decrease in the paired pulse ratio suggested a presynaptic site of TEA action. Kv3.1 and Kv3.2 were present in the terminals of these interneurones, and double knockout of these channels resulted in an increase in GABA release but also a removal of the effect of TEA on GABA. Perhaps the ability of Kv3 channels to restrict the influx of calcium is a subtle property important at subcellular sites where local calcium release is vital for Ca^{2+}-activated channel opening or vesicle fusion.

1.3.9 Use of knock out models determines Kv3 channel function

The knockout of Kv3 channels further elucidates the function of the Kv3 protein at the whole organism level, linking channel function and to a behavioural phenotype. Single null mutations (-/-) in either Kv3.1 or Kv3.3 produce only a subtle phenotype. Kv3.1 knockout display an increase in ambulatory movement and reduction in sleep (Espinosa et al. 2004), this result perhaps highlighting a role of these Kv3 channels in the sleep or circadian cycle. Furthermore, Itri et al. (2005) found higher expression levels of Kv3.1b and Kv3.2 in the suprachiasmatic nucleus (SCN) during the day. They also observed a peak in delayed rectifier currents during this period and pharmacological blockade using 1mM TEA prevented a rhythmical firing rate, further implicating Kv3 channels in the regulation of the circadian cycle and perhaps a role in rhythm maintenance or generation. Kv3.1 and Kv3.3 double knockout produced severe motor deficits, ataxia and hypersensitivity to alcohol (Espinosa et al. 2001; Hurlock et al. 2009; Matsukawa et al. 2003). These severe motor defects might be attributable to the expression, or lack thereof in double knockout animals, of Kv3 channels within circuits at both the level of the spinal cord and that of the

cerebellum. Interestingly, mice lacking Kv3.2 were unable to produce spike doublets in neocortical interneurones and thus synchronise gamma oscillations over large distances (Harvey et al. 2012).

1.3.10 Disorders that involve changes in Kv3 channels

Dysregulation or mutations in Kv3 channels can result in pathology. For example, reduction in Kv3 channels have been implicated in ageing in the auditory brainstem, (Zettel et al, 2007) and in post-mortem brains of schizophrenia patients compared to control patients (Yanagi et al 2014). Concurrently, phencyclidine based animal models of schizophrenia also report reduced Kv3 expression levels (Pratt et al, 2008). A de novo heterozygous dominant negative mutation in the gene for Kv3.1 (KCNC1) is implicated in myoclonus epilepsy, where there is a loss of function of the Kv3.1 channel, which the authors hypothesised could affect the proper functioning of Kv3-positive inhibitory interneurones (Oliver et al, 2017, Muona et al, 2015). Many mutations in Kv3.3 channels can lead to spinocerebellar ataxia (Zhang and Kaczmarek, 2016). These mutations result in voltage dependence shifts and loss of function of the channel and the condition primarily affects the cerebellum, which degenerates as the condition progresses. In spinal cord injury, Kv3.4 channels are down-regulated in dorsal root ganglia (DRG), reducing the repolarisation efficiency of DRG neurones therefore causing peripheral hypersensitivity (Ritter et al, 2015).

1.3.11 Summary of Kv3 channels

In summary it is evident that the expression pattern and modulation of Kv3 channels is crucial for determining the role that these channels play within the CNS and the phenotype of neurons in which they are expressed. Subcellularly the expression of Kv3 is heavily regulated by a number of factors such as N-glycosylation, alternative splicing, ancillary subunit interaction and heteromerisation. The subcellular expression also appears to determine the channels role; expression in dendrites may confer regulation against back propagation (Martina, Yao and Bean 2003), expression in axons ensure propagation of fast firing, and terminal expression regulates the release of neurotransmitter from the presynaptic to the postsynaptic membrane. The regional variation of Kv3 channels, often overlapping, impresses a complex picture in which Kv3 channels facilitate a variety of functions. From the non-excitable T lymphocyte to the fast spiking interneuron, to the oscillation of cortical networks and the tonotopic representation of frequency, Kv3 channels prove to be very versatile in adapting their response to a variety of stimuli. For further extensive review see Kaczmarek and Zhang (2017).

Kv3 expression has been established in the spinal cord and described here, however, the functional involvement of these channels in spinal cord circuitry and reflexes arising from the spinal cord is unexplored. The next sections explore the anatomy of the spinal cord, properties of cells in the spinal cord and the spinobulbospinal micturition reflex.

1.4 Spinal cord

1.4.1 Anatomy

The spinal cord is the longest part of the central nervous system and can be segmented into several levels, cervical, thoracic, lumbar and sacral, with each level containing circuitry generally responsible for specific regions of the body. Each level is anatomically distinct, however several features give a general anatomical description. A transverse view of the spinal cord reveals a butterfly-shaped grey matter enclosed by white matter. The white matter contains descending and ascending axonal tracts and the grey matter contains cell bodies. This view is assigned anatomical markers dorsal, ventral, lateral and medial **(see Fig 1.5).** Sensory information enters the spinal cord via a dorsal axonal root of afferents and commanded information is outputted from the spinal cord via a ventral axonal root. Many afferents terminate in the dorsal horn, a region heavily involved in sensory processing or in some cases directly onto motoneurones in the ventral horn. Like the rest of the CNS, the spinal cord contains glia and neurones, and the neurones are broadly divided into two classes; interneurones and motoneurones. Another broad division of motoneurones further separates them into two types, autonomic and somatic, with the former outputting sympathetic or parasympathetic drive to smooth and cardiac muscles of organs and the latter outputting drive to striated muscles of limbs, the trunk and sphincters. Axons leaving the spinal cord arise from these two sources; autonomic neurones in the lateral horn and somatic neurones in the ventral horn.

1.4.2 Properties of spinal interneurones

There are many distinct types of interneurones in the spinal cord. Some are excitatory meaning that they release glutamate and some are inhibitory meaning that they release GABA or glycine or both. Some project segmentally and commisurally (from one side to the other) within a spinal level and some project large distances to other spinal levels or supraspinal targets (Cote et al, 2018). Recent single cell transcriptomics of P12 spinal cords identified 10 types of GABAergic interneurones (some were also glycinergic) and 15 types of glutamatergic interneurones (Rosenberg et al, 2018).

In the dorsal horn, interneurones have been well characterised. Some excitatory interneurones, known as projection neurones, project supraspinally to contralateral postsynaptic targets but with local axon collaterals in the spinal cord. Other classes of excitatory interneurones include cone shaped vertical cells distinguished by ventral facing dendrites and radial cells with short dendrites. Islet cells with long rostro-caudal dendrites are inhibitory whereas dorsal horn central cells, similar to islet cells but with smaller dendrites, appear to be a heterogenous mix of both excitatory and inhibitory populations (Todd AJ, 2017). Whilst these morphological characterisations are useful, there is however evident overlap in the neurochemistry (neurotransmitters, calcium binding proteins and neuropeptides) of these morphological classes that makes specific populations difficult to identify and study. This appears to be a classification problem not confined to the dorsal horn in the spinal cord.

Ventral interneurones have typically been studied in the context of involvement in locomotion. They include group 1a interneurones in lamina VII that inhibit

antagonist motoneurones, Renshaw cells in lamina VII that are activated by motoneurones to inhibit motoneurones via recurrent inhibition, group 1b interneurones in lamina VII and lamina VI that can excite or inhibit motoneurones, and group II interneurones either in lamina VI-VII or IV-V that are usually part of polysynaptic pathways to excite or inhibit motoneurones. In addition, a population(s) of inhibitory interneurones mediate presynaptic inhibition of glutamatergic afferents that input directly onto motoneurones (primary afferent depolarisation) (Cote et al, 2018). Whilst these interneurones described aren't explicitly related to the aims of the thesis, they have been described to build a picture of the array of interneuronal connections between dorsal and ventral pathways in the spinal cord.

However, it is often the case that the spinal cord is reduced to a sensory dorsal half and motor ventral half with little consideration for intermediate zones, such as laminae V-VII, which are important in other vital functions such as autonomic outflow. Of relevance here in this intermediate zone is the expression of Kv3.1b in spinal interneurones antecedent to autonomic motoneurones (Deuchars et al., 2001; Brooke et al., 2002). Interneurones in the vicinity of autonomic motoneurones fire relatively fast (with an instantaneous firing frequency of 96 Hz, measured at room temperature in spinal cord slices) with brief action potentials (4.16 ms) and short afterhyperpolarisation durations (110 ms). Interestingly, although there are many types of excitatory interneurones in the spinal cord, Kv3 channels have only so far been associated with inhibitory populations in ventral and dorsal regions; Kv3 subunits are expressed in neurones that fire at high frequency such as Renshaw cells that

inhibit motoneurone output (Song et al, 2006) and inhibitory dorsal horn interneurones that gate sensory input (Nowak et al, 2011). As Kv3-positive interneurones reside in populations important in sensory, autonomic and somatic modulation, it is possible that modulating Kv3 channels could influence neuronal activity, and be important in the final output of functions and reflexes arising from the spinal cord. Of particular interest to us here is the circuitry that underpins bladder control, namely the micturition reflex.

1.4.3 The micturition reflex

The micturition reflex in mouse is of interest because the murine bladder receives both somatic and autonomic (parasympathetic and sympathetic) motor innervation from the spinal cord that in a coordinated interplay with afferent feedback from the bladder, and a supraspinal micturition centre (Barrington's nucleus) in the pons, directs the switch from continence (retention of urine) to effective voiding of urine (micturition) (**Fig.1.7)**. During bladder filling, the parasympathetic motoneurones (PGN) are inhibited and the somatic external urethral sphincter motoneurones (DLN) and sympathetic motoneurones (SPN) are excited, resulting in a relaxed detrusor, closed sphincter and continent bladder (Fowler et al., 2008). Conversely, during voiding, excitation is suppressed and inhibition removed, resulting in a contracting detrusor, relaxed sphincter and voiding bladder. This reflex involves several types of interneurone, both excitatory and inhibitory, to mediate segmental bladder afferent and pontine descending input, subsequently relaying this information to the motoneurones of the DLN, PGN and SPN, in the switch from continence to micturition (Shefchyk, 2001).

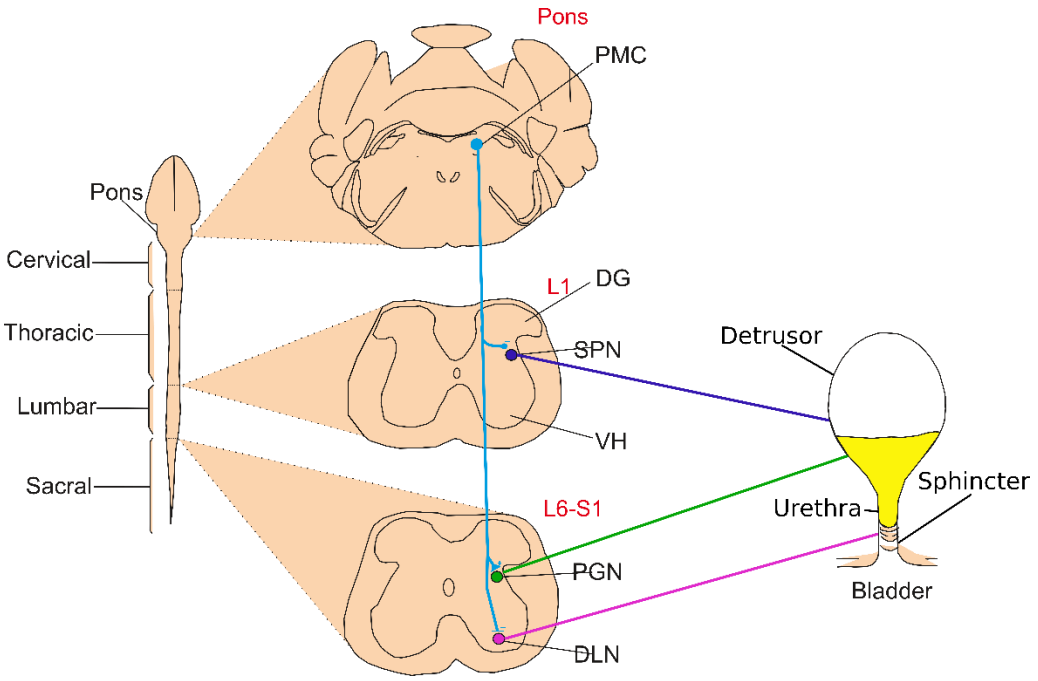


Figure 1.7. The efferent pathways in the micturition reflex emerge from the lumbosacral spinal cord.

The pontine micturition centre (PMC), stimulates the switch from continence to micturition by activating parasympathetic preganglionic and dorsolateral nucleus (DLN) in motoneurones, whilst inhibiting sympathetic preganglionic motoneurones. Sympathetic drive to relax the detrusor muscle of the bladder arises from the sympathetic preganglionic motoneurones (SPN) in the intermediolateral aspect of L1 of the spinal cord. Parasympathetic drive to contract the detrusor muscle and relax the urethra originates from the parasympathetic preganglionic motoneurones in the intermediolateral aspect of L6-S1 of the cord. Tonic contraction of the external urethral sphincter (EUS) during continence is provided by somatic motoneurones originating from the dorsolateral nucleus in the ventral horn (VH) of L6-S1 spinal cord.

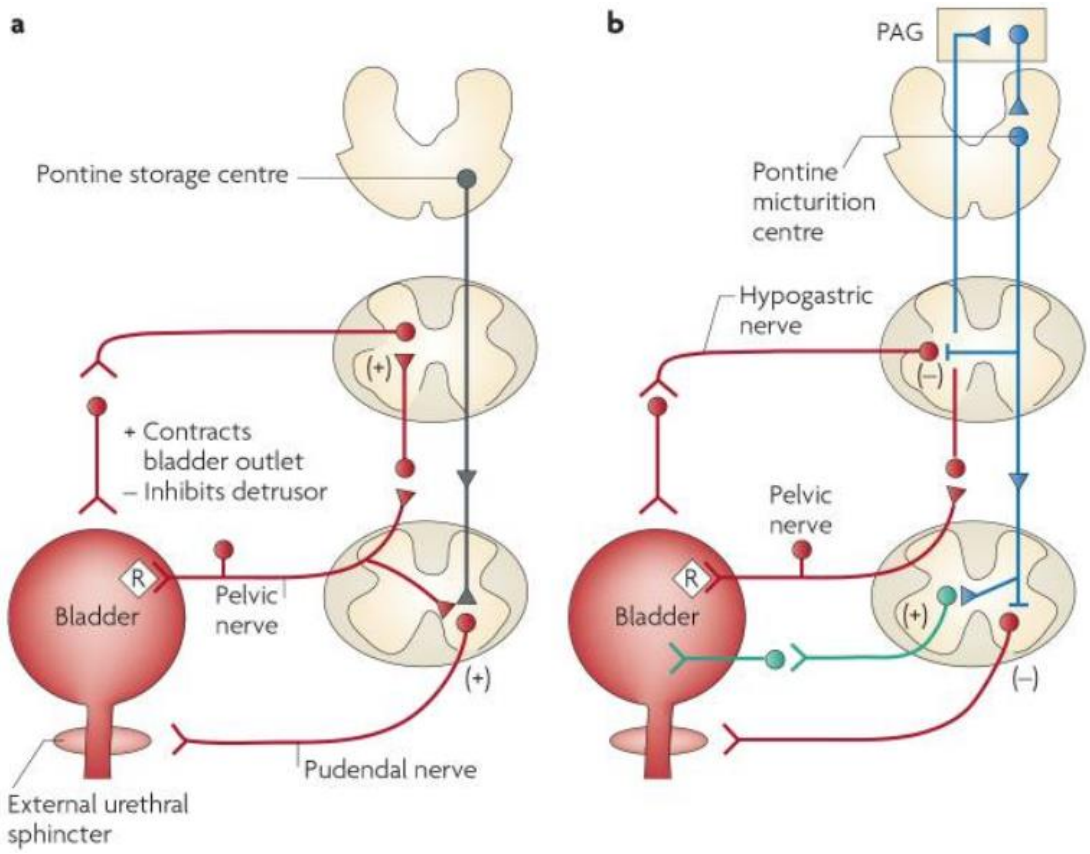


Figure 1.8. The circuits that control the switch from continence to micturition (adapted from Folwer et al, 2008)

A) Continence is maintained by the guarding reflex, Afferent firing in response to bladder filling excites sympathetic preganglionic motoneurones and sphincter motoneurones to inhibit contraction of the detrusor and stimulate contraction of the external urethral sphincter, respectively. B) The micturition reflex is initiated by suprathreshold afferent activity during bladder distension. Afferent feedback proceeds rostrally to the periaqueductal grey matter and pontine micturition centre. Descending input from these sites stimulates parasympathetic contraction of the bladder detrusor as well as inhibition of sympathetic and somatic outflows that normally maintain continence.

The urothelium and interstitial cells of the bladder offer some local control of the detrusor smooth muscle in response to distension, through the release of chemicals such as ATP, nitric oxide and acetylcholine (Fowler et al, 2008). In addition to local paracrine control, these secretions also stimulate bladder peripheral afferent endings to invoke central reflexes such as the guarding reflex during continence and the micturition reflex during voiding. Bladder afferents arise from the bladder wall and striated sphincter muscle, have their cell bodies in the dorsal root ganglia (DRG) and terminate with an overlapping pattern superficially in the dorsal horn, in the vicinity of autonomic motoneurones and dorsal to the central canal in the dorsal commissure. Afferents that respond to mechanical stimuli such as bladder distension are myelinated Aδ fibres whereas those that are quiescent during distension but respond to noxious stimuli are unmyelinated C fibres (Fowler et al, 2008). Interneurones involved in bladder control are often found where these afferents terminate and where autonomic and somatic motoneuronal dendrites arborise **(Fig. 1.8)**.

1.4.4 Interneuronal networks

During the guarding reflex, output from the parasympathetic preganglionic motoneurones is low, possibly due to indirect recurrent inhibition of the preganglionic neurones by segmental interneurones (de Groat et al, 1976). However, the sphincter motoneurones are tonically excited by polysynaptic interneuronal and afferent feedback to provide tonic contraction to the sphincter muscle (Fedirchuk et al, 1992). In the switch from continence to voiding, it is postulated that descending signals from the pons activate

excitatory interneurones which then in turn activate parasympathetic preganglionic motoneurones, increasing the parasympathetic outflow to the bladder and contracting the detrusor muscle of the bladder (de Groat et al, 1982, Araki and deGroat, 1996, 1997). In addition, interneurones located dorsal to the central canal that receive pontine input have been shown to decrease urethral pressure, possibly via direct inhibition of the sphincter motoneurones (Blok et al, 1998, Keller et al, 2018) or via presynaptic inhibition of glutamatergic afferent fibres and interneurones (Shefchyk et al, 1998).

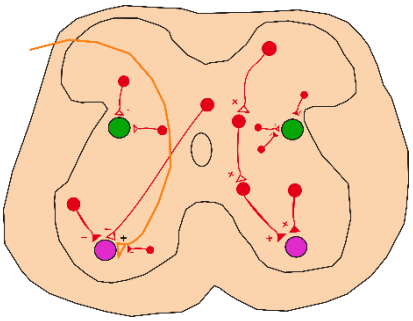


Figure 1.9. A schematic of the interneuronal populations at L6-S1 described above and in Shefchyk (2001)

The location of interneurones described in Shefchyk (2001) in the micturition reflex. Excitatory and inhibitory interneurones are in red, afferent mono-synaptic input is in orange, parasympathetic preganglionic nucleus is in green and the dorsolateral nucleus is in pink.

1.4.5 Properties of spinal motoneurones

The micturition reflex is complex in its use of both autonomic and somatic spinal motoneurones. This section describes briefly some important aspects of these spinal motoneurones. Autonomic preganglionic motoneurones are predominantly located in the intermediolateral aspect or lamina V-VII of the gray matter of the spinal cord, although three other smaller autonomic pools

exist in the nucleus intermediolateralis thoracolumbalis pars funicularis, intercalated nucleus, and central autonomic area (Deuchars and Lall, 2015). These autonomic motoneurones are the final autonomic outflow from the spinal cord, however, as the name preganglionic implies, their targets are many cell bodies in the corresponding peripheral autonomic ganglia where it is likely that further signal processsing and integration occurs before reaching the target tissue. In the micturition reflex, sympathetic motoneurones are located in L1 segments and parasympathetic motoneurones at L6-S1. The general properties and inputs of sympathetic motoneurones have been well reviewed (Deuchars and Lall, 2015, Llewellyn-Smith, 2009). SPNs are not thought to receive direct afferent input, instead sensory input is fed through interneurones that input onto SPNs. In addition to local connections, SPNs receive serotonergic and glutamatergic descending input from raphe nuclei and direct innervation from the pontine micturition centre (Barringtons nucleus). Biophysically, SPNs have been classified by absence of an Ih, broad action potentials (~ 10 ms) and steady state firing rates around 20 Hz at room temperature (Deuchars 2001). Sacral parasympathetic motoneurones have been less well characterised and recent controversy surrounds whether these neurones are genetically actually parasympathetic or sympathetic (Espinosa-Medina et al, 2016, 2018, Horn et al, 2018). These autonomic motoneurones are conventionally considered a simple final outflow of the autonomic system but one of the most interesting and relatively unexplored aspects of these autonomic motoneurones is the intra-spinal collateralisation of axons (Deuchars and Lall, 2015, Morgan et al, 1991). This collateralisation is suggestive of a much more complex role in intra-spinal signalling and recurrent feedback.

The last motoneurones to consider in the micturition reflex are the somatic motoneurones in the dorsolateral nucleus of L6-S1 spinal cord that innervate the urethral sphincter. Somatic motoneurones are typically but broadly classified into two types, α-motoneurones and γ-motoneurones, however, DLN motoneurones are distinctly smaller and packed closer than classical somatic motoneurones (Onufrowicz, 1899). Akin to conventional somatic motoneurones, DLN Motoneurones, in cat, have recurrent axon collaterals (Sasaki et al, 1994), although contrastingly no recurrent inhibition has been observed (Mackel et al, 1979). Speculatively, this perhaps suggests a recurrent facilitation to perhaps provide tonic activity and the maintenance of closure of the external urethral sphincter during continence. DLN motoneurones are therefore an unusually distinct population of motoneurones. Indeed, while many somatic motoneurones succumb to motoneurone disease, the human DLN equivalent, Onufs nucleus, is apparently well-preserved (Mannen et al, 1982).

One common feature, however, between autonomic preganglionic neurones and motoneurones of the DLN is the expression of connexin-36. This implies the existence of strong electro-coupling between neurones via gap-junctions, likely being most important in the generation of population-level synchronisation and rhythmogenicity (Bautista et al, 2014, Deuchars and Lall, 2015).

1.4.6 Motor innervation of the bladder

Peripherally and post-ganglionically, sympathetic outflow is usually associated with increases in physiological activity, such as cardiac output, however at the bladder the release of noradrenaline to activate β3-adrenergic receptors relaxes the bladder detrusor smooth muscle while activation of α1-adrenergic

receptors contracts the urethra (Fowler et al, 2008). Parasympathetic innervation utilising acetylcholine activates M3-muscarinic receptors to contract the bladder detrusor smooth muscle whist releasing nitric oxide to relax the urethra (Fowler et al, 2008). Somatic innervation from the sphincter motoneurones forms typical neuromuscular junctions with the striated muscle of the sphincter, releasing acetylcholine to activate nicotinic acetylcholine receptors for rapid muscle contraction (Fowler et al, 2008). When somatic and sympathetic outflows are active the bladder is continent, but when these are inhibited and parasympathetic outflow is activated the bladder voids.

1.4.7 Micturition changes with ageing

It is well documented that ageing results in incontinence and a loss of motor function (Siroky, 2004, Suskind, 2017). Perhaps these losses of function could be attributed to the degeneration of the spinal circuits that regulate and innervate these outflows. The process of ageing can result in widespread degeneration within spinal cord circuits: loss of descending input, reduced afferent feedback and reduction in the number of neurons such as SPNs (Santer et al. 2002). There could also be a reduction in the synchronisation of neuronal firing; for example, the motoneurone activity of the elderly is scattered compared with defined bursts of activity in motoneurones of young people (Monaco, Ghionzoli and Micera 2010). In the bladder, up to 40 % of men and 30 % of women over 75 years, experience overactive bladder symptoms, presenting with increased voiding frequency during the data and night (nocturia) (Milsom et al, 2000). Specifically, in the bladder, ageing manifests as an increased detrusor activity and less efficient contractility, which may be due

to age-related changes in the micturition reflex (Siroky, 2004). The process of ageing affects many physiological processes and it is important to characterise not only the changes that are happening at the level of the bladder (such as detrusor fibrosis or changes in neurotransmitter sensitivity (Suskind, 2017)) but also within the spinal cord circuitry that largely underpins the micturition reflex.

1.4.8 Research questions, hypothesis and aims

Whilst the expression of Kv3 channels has been observed in the spinal cord (described above), little is known about the level of expression in ageing. Kv3 channels have been shown to decrease with age in the auditory brainstem (Zettel et al. 2007). It is therefore possible to postulate that a similar decline could be occurring in the spinal cord and even throughout the CNS. As discussed, Kv3 channels are present in interneurones antecedent to sympathetic preganglionic neurons (SPNs). These neurons innervate many tissues and provide the final outflow of sympathetic activity onto these tissues, such as the bladder.

Throughout this introduction, I have highlighted a role for Kv3 channels in neuronal firing and synaptic transmission. It would be a logical step to investigate the involvement of Kv3 channels in this type of regulation within motor and autonomic circuits at the spinal level of the bladder reflex. In addition, using new pharmacological tools selective for Kv3 channels such as AUT1 (Autifony Therapeutics, Ltd) allows one to dissect a specific contribution to the bladder phenotype. Whilst the phenotypes of Kv3 knockouts are subtle, the effects of pharmacological modulation or of a decline of Kv3 channels with

age could be significant, especially without developmental compensation. It would therefore be important to observe whether expression levels of Kv3 channels change throughout ageing and if so what effect this produces on the output of neuronal populations in which Kv3 channels are present.

1.4.9 Aims

- To investigate if Kv3 expression changes with age within the bladder reflex and whether this correlates with loss of bladder function experienced during ageing.
- To explore the therapeutic potential of a Kv3 selective modulator, AUT1, by studying its effect on recombinant Kv3 channels, on the excitability of neurones in the spinal cord, and on bladder function in young and aged mice.

Chapter 2

2 Kv3 channels in lumbosacral spinal synapses

2.1 Introduction

2.1.1 Bladder motoneurones; the final spinal outputs to the bladder

In the circuitry involved in the micturition reflex, discussed in General Introduction, inhibitory and excitatory synapses from local interneurones and peripheral and descending inputs terminate onto somatic dorsolateral nucleus (DLN), autonomic parasympathetic preganglionic (PGN) and sympathetic preganglionic (SPN) motoneurones in the lumbo-sacral spinal cord. DLN motoneurones in the ventral horn innervate the external urethral sphincter and PGN and SPN motoneurones in the lateral gray matter innervate the smooth muscle of the bladder detrusor muscle and urethra. In a coordinated interplay with afferent feedback and descending pontine input, these different types of motoneurone mediate bladder continence and voiding; DLN motoneurones tonically contract the external urethral sphincter, PGN motoneurones contract the bladder detrusor and relax the urethra and SPN motoneurones relax the detrusor and contract the urethra. The composition of synaptic inputs that these motoneurones receive and integrate is extremely important in determining their output and effect of this output on bladder function.

2.1.2 Kv3 immunoreactivity in the vicinity of motoneuronal pools in the spinal cord

Punctate immunoreactivity of Kv3 subunits, Kv3.1b, Kv3.3 and Kv3.4, has previously been localised to pre-synaptic and post-synaptic structures in the ventral horn of the thoracic spinal cord (Brooke *et al.*, 2004, 2006). Double labelling immunohistochemistry with both excitatory and inhibitory synaptic markers indicated that Kv3 subunits were associated with both types of synapses around neurones that included motoneurones in this region. In addition, again at the thoracic level, Kv3.1b immunoreactivity was found in the vicinity of autonomic motoneurones in the lateral gray matter and in interneurones antecedent to these autonomic motoneurones (Deuchars *et al.*, 2001; Brooke *et al.*, 2002). These findings are indicative of a role of Kv3 channels in synaptic transmission onto spinal motoneurones, however, to date no analysis has been conducted on the role of Kv3 subunits in pre-synaptic structures at the lumbo-sacral level of bladder control where these motoneuronal outputs are heavily coordinated.

2.1.3 Kv3 channels constrain neurotransmitter release

At synapses in other regions of the CNS, Kv3 channels constrain neurotransmitter release from the presynaptic terminal by ensuring action potential waveforms are kept brief and limiting Ca2+ influx (see General Introduction). Therefore, it is likely that Kv3 channels play a similar role in spinal synapses, restricting the amount of neurotransmitter released to post-synaptic bladder motoneurones. The implications of Kv3 channels in spinal synapses in bladder circuity are three fold. Firstly synaptic transmission where delayed rectifier Kv3 subunits predominate would be reliable during high

frequency rates of discharge. However, if rapidly inactivating Kv3 subunits are expressed, then synaptic transmission would be potentiated in an activity dependent manner as inactivation proceeds. And finally pharmacological modulation via blockade or gating modification of Kv3 channels should affect transmission and therefore the integration/output of motoneurones in the micturition reflex.

2.1.4 The context of Kv3 expression is important and changes with age

The context of Kv3 channel expression has clear functional relevance; at a subcellular level, pre-synaptic expression regulates neurotransmitter release and somatic expression facilitates high frequency firing, while at a multi-cellular level, in the auditory brainstem, channel expression follows a frequency mapping tonotopic gradient (Parameshwaran et al, 2001, Li et al, 2001). Curiously, during the ageing process of the auditory brainstem, Kv3 channel expression decreased, the gradient degraded and functional consequences were observed, specifically, a decline in medial olivocochlear efferent activity (von Hehn et al., 2004; Zettel et al., 2007). Therefore, in studying the synaptic expression of Kv3 channels, it is important to consider the subunit expression through the ageing process. Furthermore, as previously eluded to, Kv3 channels are predominantly associated with inhibitory interneurones in the brain, but appear to be more promiscuous in their association with excitatory and inhibitory neurones in the spinal cord. It is therefore important to understand the context of Kv3 channel expression, examining both excitatory and inhibitory synapses in this investigation into the role of Kv3 channels in bladder circuitry.

2.1.5 Hypothesis and Aims

Hypothesis: that Kv3 channels are expressed in synapses in bladder circuitry and that their levels and expression patterns change with age.

Aims:

- **To investigate Kv3 and synaptic immunoreactivity in the lumbosacral spinal cord of 3 month and 28 month mice.**
- **To investigate the functional role of Kv3 channels in spinal synapses**

2.2 Materials and Methods

All procedures performed in accordance with UK Animals (Scientific Procedures) Act 1986 and ethical standards set out by the University of Leeds Ethical Review Committee. Every effort was made to minimize the number of animals used and their suffering.

2.2.1 Spinal cord tissue preparation

C57bl6 mice (p10-21) from Central Biomedical Services at the University of Leeds were anaethetised by intraperitoneal (i.p) injection of sodium pentobarbitone (Euthanal, 60 mg/kg). Upon complete loss of pedal withdrawal a transverse laparotomy was carried out to remove the ventral ribs and expose the heart. The right atrium was incised and 20 ml of sucrose artificial cerebrospinal fluid (sucrose aCSF) oxygenated on ice was perfused through the left ventricle and into the circulation system. The mouse was decapitated, the

skin removed and a dorsal laminectomy carried out to expose the spinal cord with removal requiring cutting of the rootlets attached to the cord.

Upon removal, the spinal cord was placed in a petri-dish containing ice-cold sucrose aCSF (see Table) under a dissecting microscope (SM2 2B, Nikon) and the meninges that ensheathe the cord were removed with fine forceps. Lumbo-sacral segments of the spinal cord were set in 3 % agar in aCSF, mounted against a 4 % block of agar for stability using superglue and sectioned in a bath of oxygenated ice-cold sucrose aCSF using an Integraslice 7550 PSDS (Campden Instruments, UK) microtome. Transverse sections were cut at 250-300 μm and incubated in an oxygenated holding chamber containing aCSF (see Table) and allowed to recover for an hour before recording.

Substance	**Sucrose aCSF (mM)**	**aCSF (mM)**
Sucrose	217	
NaCl		124
NaHCO3	26	26
KCl	3	3
MgSO4.7H20	2	2
NaH2PO4	2.5	2.5
Glucose	10	10
CaCl2	1	2

Table 2.1 Extracellular composition

2.2.2 Current clamp

Recordings were carried out at room temperature (18-24 °C). Slices were transferred to and immobilised in an incubation chamber perfused with oxygenated aCSF at a rate of 3-5 ml/minute from a flask above.

Thin walled borosilicate glass microelectrodes (inner diameter 0.94 mm, outer diameter 1.2 mm) were fabricated using a Sutter P97 micropipette puller (Sutter Instruments, USA) with resistances of 5-9 MΩ. The recording and bath electrode used a silver/silver chloride (Ag/AgCl2) interface and connected to a CV-4 1/100U headstage (Axon Instruments, USA). Microelectrodes were filled with an intracellular solution composed of; K gluconate, 110 mM; EGTA, 11 mM; MgCl2, 2mM; CaCl2, 0.1 mM; HEPES, 10 mM; Na2ATP, 2 mM; Na2GTP, 0.3 mM and 0.5% Neurobiotin (Vector Laboratories, USA).

Recordings were obtained at 5-20 kHz, filtered through a Bessel low pass filter at 2-5kHz using an Axopatch 1D amplifier (Axon Instruments, USA), a humbug to eliminate mains noise (Quest Scientific, via Digitimer, UK), and were digitised using a Digidata 1322A (Axon Instruments, USA) and recorded in pClamp9 software. A Master 8 was used to define the duration and timing of current steps set by the amplifier.

An upright microscope (Olympus BX50W1), camera (QImaging Rolera-XR, QImaging, Canada) and QCapture software (QImaging, Canada) was used to visualise the spinal cord section and centre the stage over the region of interest e.g. the lateral region of the lumbo-sacral spinal cord sections. Infra-red differential interference contrast (DIC) microscopy was used to penetrate

deeper in to the section and to better visualise the spinal neurones. Cells were selected in the region of autonomic motoneurone nuclei based on their appearance (not swollen or round) and depth (not at the surface) in order to obtain recordings from healthy cells. In this region, interneurones fire at faster frequencies than autonomic motoneurones (Deuchars 2001). The maximal firing frequency of all cells included in the analysis was less than 20Hz.

In solution and current clamp mode, the electrode was offset to zero and the voltage response to a -250 pA step was offset using the series resistance dial, essentially reflecting electrode resistance. A small amount of positive pressure was applied to the pipette as it was lowered to the surface of the section. In voltage clamp and track mode, a -25 mV step was applied to the pipette as it was lowered to the cell of interest. The positive pressure was released upon identification of a small dimple in the membrane of the cell to form a GΩ seal. The amplifier was returned to current clamp mode and using a -50 pA step and brief suction the seal was ruptured to provide whole cell access.

In whole cell configuration, neurones were characterised by long (1 second) hyperpolarising and depolarising current steps from a holding potential of -70 mV. Excitatory post-synaptic potentials (EPSPs) were evoked by a brief pulse of ~ 8V using a bipolar electrode positioned in the lateral white matter where descending tracts are located. During EPSP stimulation, neurones were held at -100 mV to prevent activation of postsynaptic TEA sensitive cells. 0.5 mM TEA was bath applied for 3 minutes and washed off for 10 minutes. EPSPs and characterising sweeps were recorded in control, TEA and wash conditions. Cells with single component EPSPs were selected for analysis. The amplitude,

duration and latency from stimulation for each EPSP (>10 EPSP for each condition) were measured.

2.2.3 Immunohistochemistry

Somatic and autonomic motoneurones were labelled by intraperitoneal injection of hydroxystilbamidine (Fluorogold, FG (Hydroxystilbamidine, Abcam), 0.1 ml of 1 % i.p. in H_2O) 1d before perfusion. Female wild-type C57BL/6 (3 month old, n=3; 28 month old, n=3) were anesthetized with intraperitoneal pentobarbitone sodium (Sagatal, 60 mg/kg) and perfused transcardially with 4% paraformaldehyde (PFA) in 0.1 M phosphate buffer (PB), pH 7.4. Brains and spinal cords were removed and post-fixed overnight in 4% PFA. After fixation, spinal levels L1, L6, S1 were dissected and incubated in PBS 0.1M containing 30 % sucrose until the tissue sank to the bottom before being embedded and frozen in Surgipath FSC 22 Clear Frozen Section Compound (Leica) freezing medium on dry ice. 20 µm sections were cut using a Leica CM1850 cryostat cooled to approximately -15 ˚C and mounted onto Superfrost plus (Menzel-Glaser, Thermo Scientific) slides for each condition (~8) such that any two sections for one condition were separated by at least 120 µm as the tissue was sequentially cut. This was to avoid analysis of the same neurones.

Sections were washed 3 times in PBS, incubated in 10 mM sodium citrate at 80°C for 20 minutes for antigen retrieval, washed a further 3 times in PBS and blocked and permeabilised for 1 hour in 5 % goat and donkey serum in PB (0.3 % Triton X-100). All primary antibodies were incubated overnight in PB (0.3% Triton X-100), washed in PBS and incubated for 1 hour for directly conjugated secondary antibodies, for 2 hours for biotinylated secondary antibodies and for

30 minutes for streptavidin to avoid endogenous biotin labelling (Table 2.2 for antibody concentrations and secondary detection).

Target	Supplier	Species Raised in	Dilution	Secondary Detection	Cat. No	Reference
Kv3.1b	Neuromabs/ Antibodies Inc	Mouse	1:100	Biotinylated α- mouse IGg1 (Invitrogen A10519), Streptavidin Alexa 555 (Invitrogen)	75-041	Soares et al 2017
Kv3.3	Neuromabs/ Antibodies Inc	Mouse	1:100	Biotinylated α- mouse IGg1 (Invitrogen A10519), Streptavidin Alexa 555 (Invitrogen)	75-354	Soares et al 2017
GlyT2 (Glycine Transporter)	Synaptic Systems	Rabbit	1:2000	Donkey α-Rabbit Alexa 488 (Life Technologies, A21206)	272 003	Nelson et al 1995
VGAT (Vesicular GABA Transporter)	Synaptic Systems	Rabbit	1:2000	Donkey α-Rabbit Alexa 488 (Life Technologies, A21206)	131 003	Tozuka et al 2005
VGluT2 (Vesicular Glutamate Transporter)	Synaptic Systems	Rabbit	1:2000	Donkey α-Rabbit Alexa 488 (Life Technologies, A21206)	135 403	Zhu et al 2018
ChAT (Choline acteyltransferase)	Abcam	Goat	1:1000	Donkey α-Goat Alexa 488 (Life Technologies, A21206)	Ab18736	Wang et al 2017

Table 2.2 Antibodies and concentrations used

2.2.4 Confocal microscopy

Confocal data were acquired using a Zeiss LSM880 Upright microscope and 63x oil immersion lens. An overview tile scan of Fluorogold in a spinal section was taken to identify motoneurones of interest; the DLN in the ventral horn of L6-S1, the PGN in the lateral aspect of L6-S1 and the SPN in the lateral aspect of L1. Images of motoneurones were acquired with the nucleus in view. DIC was used to identify the cell outline and determine a visible nucleus whilst a super-resolution Airyscan image was taken of the motoneurone with the Alexa 555 and Alexa 488 fluorophores being stimulated separately and emissions collected with appropriate bandpass filters for the fluorophores. Due to an unexpected bleed-through of the Fluorogold spectra into the 488 channel images, a lambda stack was also taken, spectrally unmixed by a linear unmixing algorithm performed in Zen, which calculates the concentration of a discrete range of wavelengths in the intensity of each individual pixel. With the Fluorogold emission removed, these spectrally unmixed images were then analysed for co-localisation. At least 3 sections at each level and for each condition with 3 motoneurones per section were analysed for each animal.

2.2.5 Analysis

The motoneurone perimeter was traced in ImageJ to create a region of interest (ROI). A 3 μm band around the cell was created from 2 μm outside the perimeter to1 μm inside the perimeter. This formed a ROI from which synaptic immunoreactivity and Kv3 puncta in very close apposition with the cell membrane could be segmented **(Fig. 2.1)**. Synaptic immunoreactivity, referred to as boutons herein, and Kv3 puncta were segmented by application of a

Phansalkar local threshold method, a 'Make Binary' function and an 'Analyse Particles' function in ImageJ, where parameters specifying boutons had to be between 0.5 µm and 5 µm in area and Kv3 puncta between 0.1 and 5 µm in area, fall within the ROI but be excluded if located on the edge of the ROI. For object-based co-localisation, co-localisation was defined as the centre of a Kv3 punctum coinciding with the area of a bouton. This was performed using the JaCOP plugin (Cordelieres et al, 2006) on ImageJ and reported as the percentage of co-localised boutons and as the percentage of co-localised puncta.

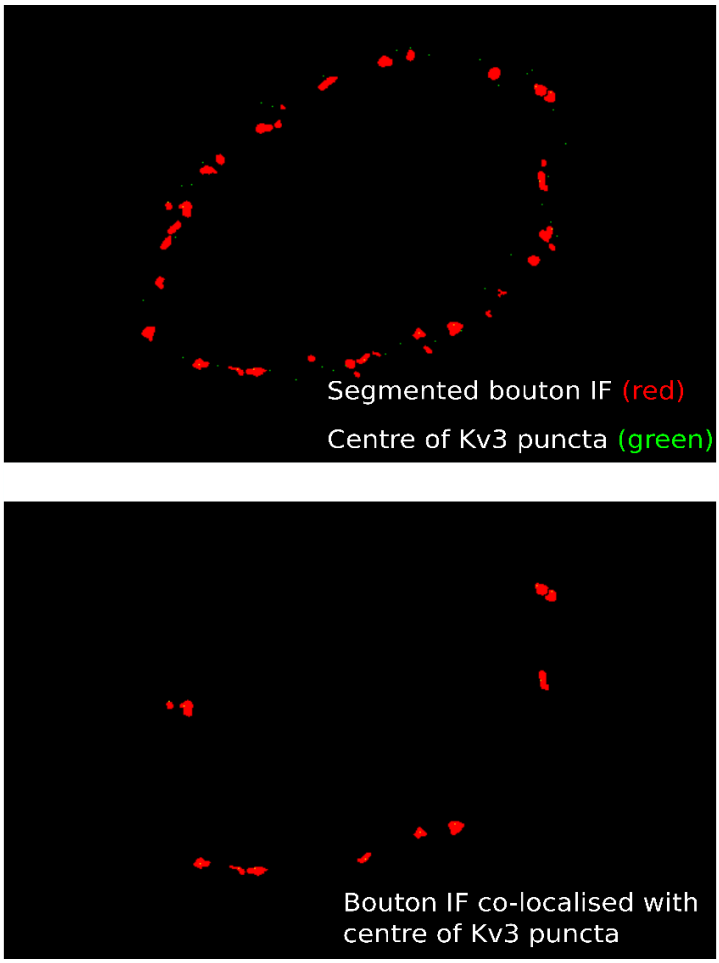


Figure 2.1. Object overlap o-localisation analysis.

Segmented bouton immunofluorescence and the centre of Kv3 puncta. Object-overlap based co-localisation revealed number of boutons containing Kv3 puncta.

2.2.6 Statistics

Statistical analysis was carried out in R statistical software. All groups were analysed for homogeneity of variance and normality using Levene's test and Shapiro-Wilks, respectively. Repeated measures ANOVA for parametric data and Friedman tests for non-parametric data were performed on EPSP data where the groups control, TEA and wash were related. Post-hoc analysis of significant variables was performed using pairwise t-tests or Wilcox tests with p values corrected automatically using the Bonferroni method. Synaptological comparisons between young and aged mice were performed on pooled data using independent t-tests or a Wilcoxon rank sum test (an equivalent to the Mann Whitney test) and p values were corrected using the Holm method.

2.3 Results

2.3.1 Replication of data regarding Kv3 immunoreactivity from the literature

Initial immunohistochemistry sought to replicate data for Kv3 immunoreactivity published from rat thoracic spinal cord sections in mouse thoracic sections using Neuromab antibodies (where previous studies had used antibodies from Alomone, Brooke 2006). The pattern of staining observed was highly comparable (see Discussion); Kv3.1b and Kv3.3 immunofluorescence was widespread in the thoracic spinal cord and was very punctate and often ring-like, indicative of close apposition or expression at the membranes of cellular structures such as somata **(Fig. 2.2 A, B).** Double-labelling with ChAT, a marker of cholinergic neurones and motoneurones, indicated several features of Kv3 expression. Firstly, Kv3.1b and Kv3.3 ring-like immunoreactivity was absent from the intermediolateral region (IML) **(Fig. 2.2 AI, BI)** but instead was represented by small puncta close to the labelled cell bodies of autonomic motoneurones. Secondly, ring-like immunoreactivity in the central canal region was largely absent from cholinergic somata **(Fig. 2.2 AII, BII).** Reactivity in the ventral horn was punctate and intense and closely apposed (indicative of synaptic expression) to the large cholinergic cell bodies that represent the somatic motoneuronal population of the thoracic spinal cord **(Fig. 2.2 AIII, BIII).**

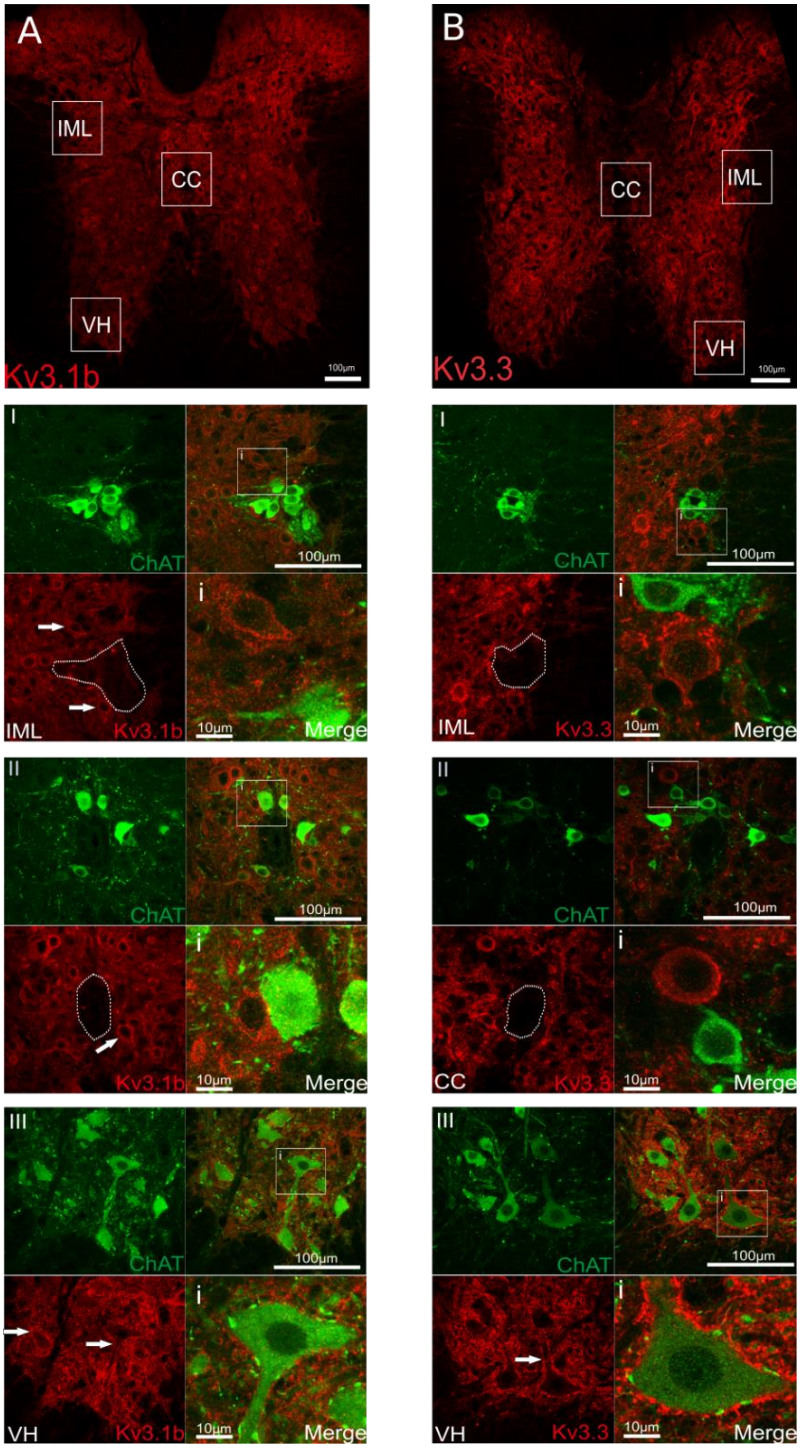


Figure 2.2. Kv3 –IF was replicated using Neuromab antibodies

A,B; *Kv3.1b (A) and Kv3.3 (B) immunoreactivity (red) in the thoracic murine spinal cord White boxes mark the intermediolateral horn (IML), the central canal (CC) and the ventral horn (VH).* ***A, BI, II, III;*** *Double labelling of Kv3 subunits with Choline-acetyl-transferase (ChAT) highlights the distribution of Kv3 immunoreactivity in relation to cholinergic neurones, predominately autonomic motoneurones in the IML and somatic motoneurones in the VH (I, II, III). White outline represents the autonomic nucleus in I and the central canal in II. White arrows indicate examples of ring-like Kv3 immunoreactivity. Magnified cholinergic neurones in relation to Kv3 immunoreactivity (i).*

2.3.2 Are Kv3 subunits in synapses onto bladder motoneurones?

Because Kv3 immunoreactivity was closely apposed to ChAT-positive motoneurones **(Fig 2.2)** in a pattern typical of synaptic expression, the focus of this investigation was on Kv3 expression in synapses onto bladder motoneurones. Triple labelling immunohistochemistry was performed to test this hypothesis that Kv3 channels were expressed in synaptic inputs onto somatic and autonomic motoneurones arising from regions of the spinal cord involved in the micturition reflex, and thus could be involved in the mediation of this reflex. Three neurones per section with three sections per mouse for three mice were analysed.

2.3.3 Kv3 co-localisation in the DLN

Motoneurone cell bodies residing in the DLN of the ventral horn of L6-S1, the PGN of the lateral aspect of L6-S1 and the SPN of the lateral aspect of L1, were traced from the periphery with Fluorogold **(Fig. 2.3-2.8).** To look at the context of Kv3 expression, markers of transporters for both excitatory (VGluT2) and inhibitory (VGAT and GlyT2) neurotransmitters found in synaptic terminals and synaptic vesicles were used (**see Table 2.2).**

Kv3.1b immunoreactivity was characterised as widely expressed small puncta distinct around cell somata and also in the neuropil of L1, L6 and S1 spinal levels **(Fig. 2.3, 2.5, 2.7)**. Kv3.3 immunoreactivity was more intense and represented visibly larger puncta around cell somata and also in fibre-like structures **(Fig. 2.4, 2.6, 2.8)**. For simplicity, the immunoreactivity for the synaptic or synaptic vesicular markers used here, is referred to as boutons. Co-localisation was defined as the centre of a Kv3 punctum occurring within the

area of a synaptic bouton. Values for this definition of co-localisation are reported as the percentage of co-localised boutons and the percentage of co-localised puncta. Essentially, this shows how many boutons had a Kv3 punctum fall within its immunoreactive area and also how many puncta fell within the immunoreactive area of a bouton (See Materials and Methods).

For DLN motoneurones, those that putatively innervate the external urethral sphincter, 24 % of GlyT2, 15 % of VGaT and 20 % of VGluT2 boutons were co-localised with Kv3.1b puncta. Only 8%, 8% and 5% of Kv3.1b puncta, however, were co-localised with a GlyT2, VGaT and VGluT2 bouton, respectively. For Kv3.3 double labelling, 17 %, 11 % and 19 % of GlyT2, VGaT and VGluT2 boutons, respectively, contained a Kv3.3 punctum. Again however, the numbers of co-localised Kv3.3 puncta were lower at 9%, 4% and 9% for GlyT2, VGaT and VGluT2 boutons. Kv3.1b and Kv3.3 can be seen to be co-localised or closely associated in high resolution images with synaptic markers GlyT2, VGaT and VGluT2 **(Fig. 2.3 and 2.4 A, B, C; i, ii, ii).**

In summary, Kv3.1 and Kv3.3 -IF co-localised with both excitatory and inhibitory synaptic markers in closely apposed to putative DLN motoneurones.

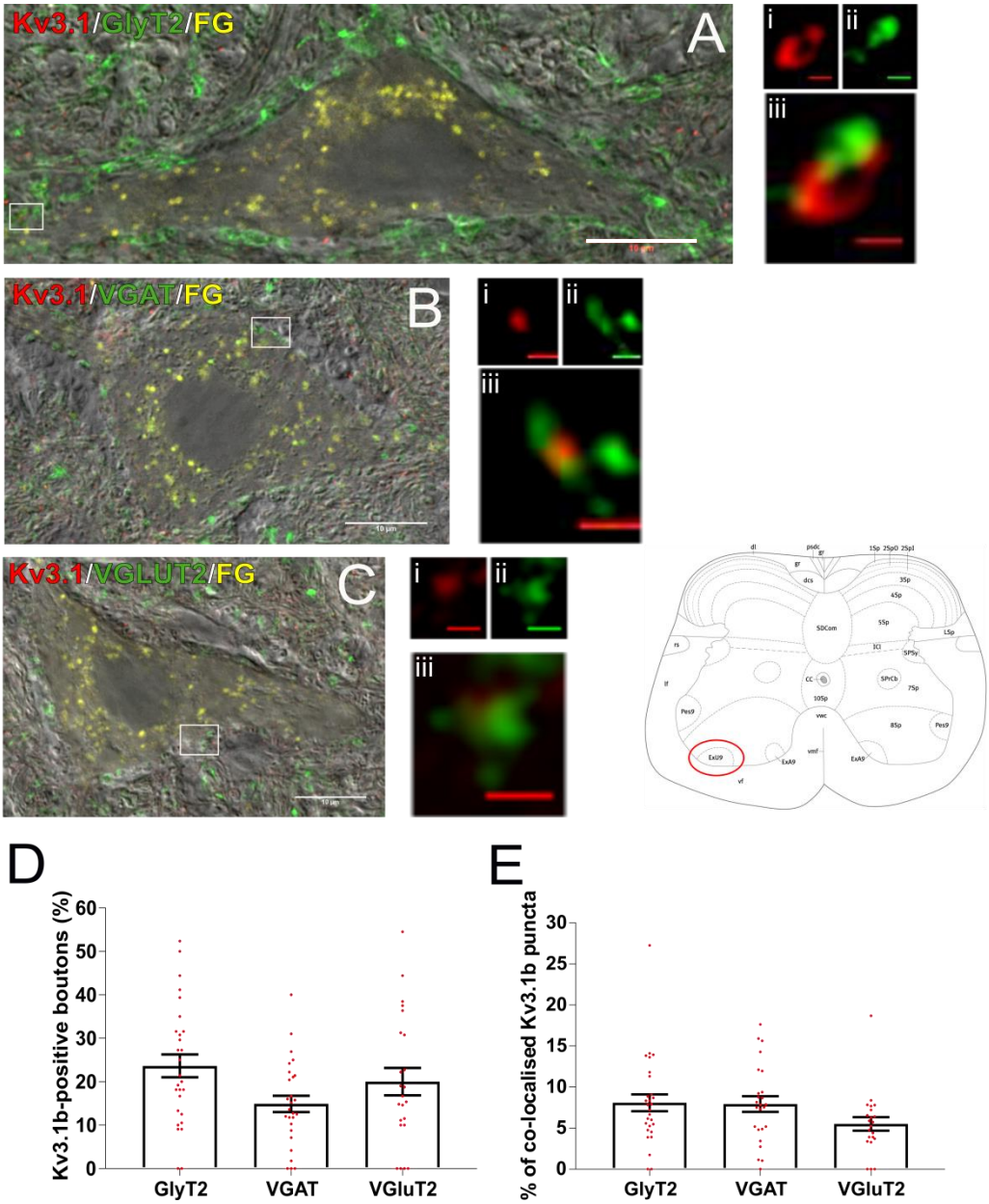


Figure 2.3. Kv3.1-IF puncta co-localised with inhibitory and excitatory synaptic markers onto DLN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.1b (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative DLN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord).* *Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii).* *Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

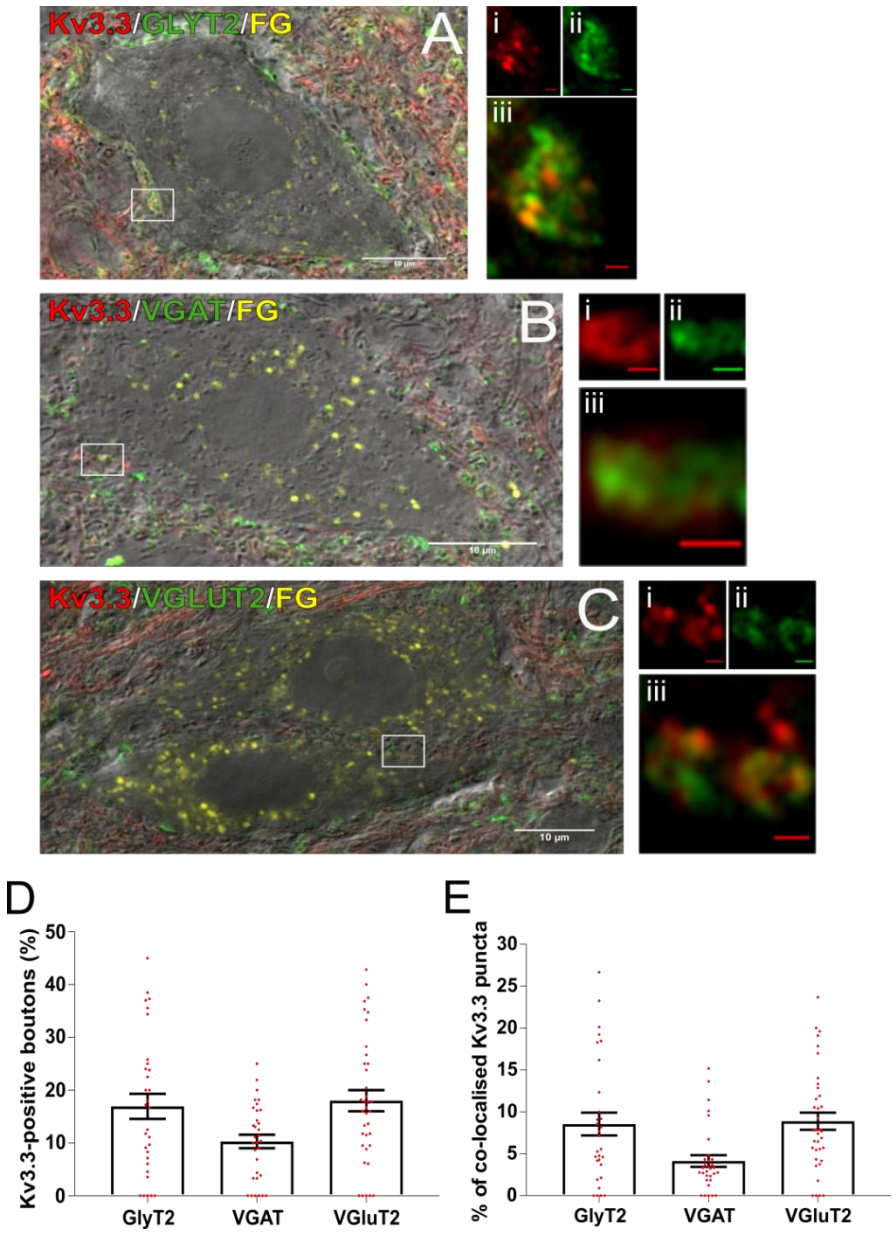


Figure 2.4. Kv3.3-IF puncta co-localised with inhibitory and excitatory synaptic markers onto DLN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.3 (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative DLN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord).* *Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii).* *Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

2.3.4 Kv3 co-localisation apposed to PGN motoneurones

For PGN motoneurones, those that putatively stimulate voiding of the bladder, 14% GlyT2, 13% VGaT and 12% VGluT2 boutons co-localised with Kv3.1b puncta whereas 6%, 3%, 4% of Kv3.1b puncta co-localised with GlyT2, VGaT and VGluT2 boutons, respectively (**Fig. 2.5).** Similarly, 11% GlyT2, 9% VGaT and 8% VGluT2 boutons were co-localised with Kv3.3 puncta whereas 6%, 2% and 4% Kv3.3 puncta were co-localised with GlyT2, VGaT and VGluT2 boutons, respectively (**Fig. 2.6)**.

In summary, Kv3.1 and Kv3.3 -IF co-localised with both excitatory and inhibitory synaptic markers in closely apposed to putative PGN motoneurones.

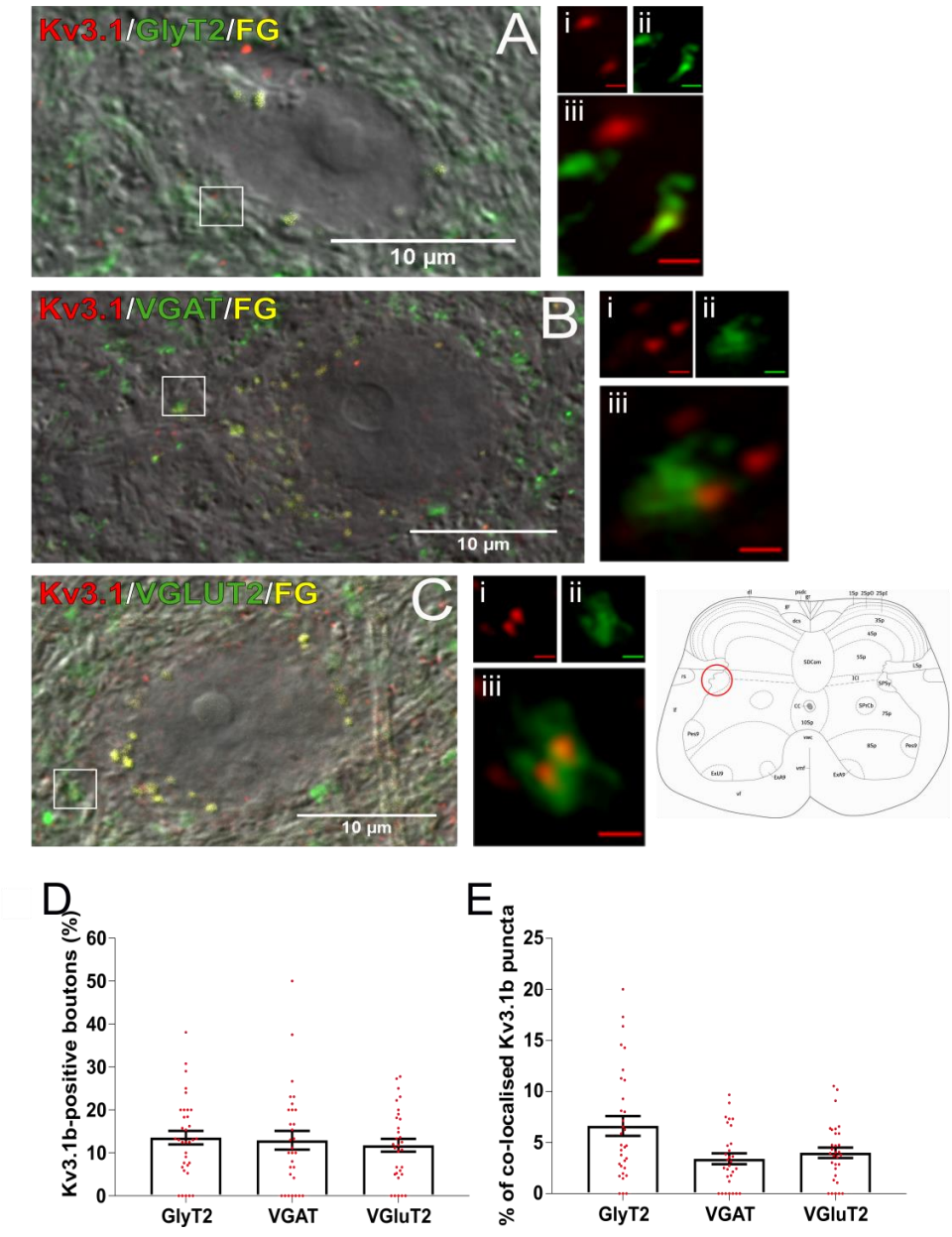


Figure 2.5. Kv3.1-IF puncta co-localised with inhibitory and excitatory synaptic markers onto PGN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.1b (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative PGN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord). Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii). Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

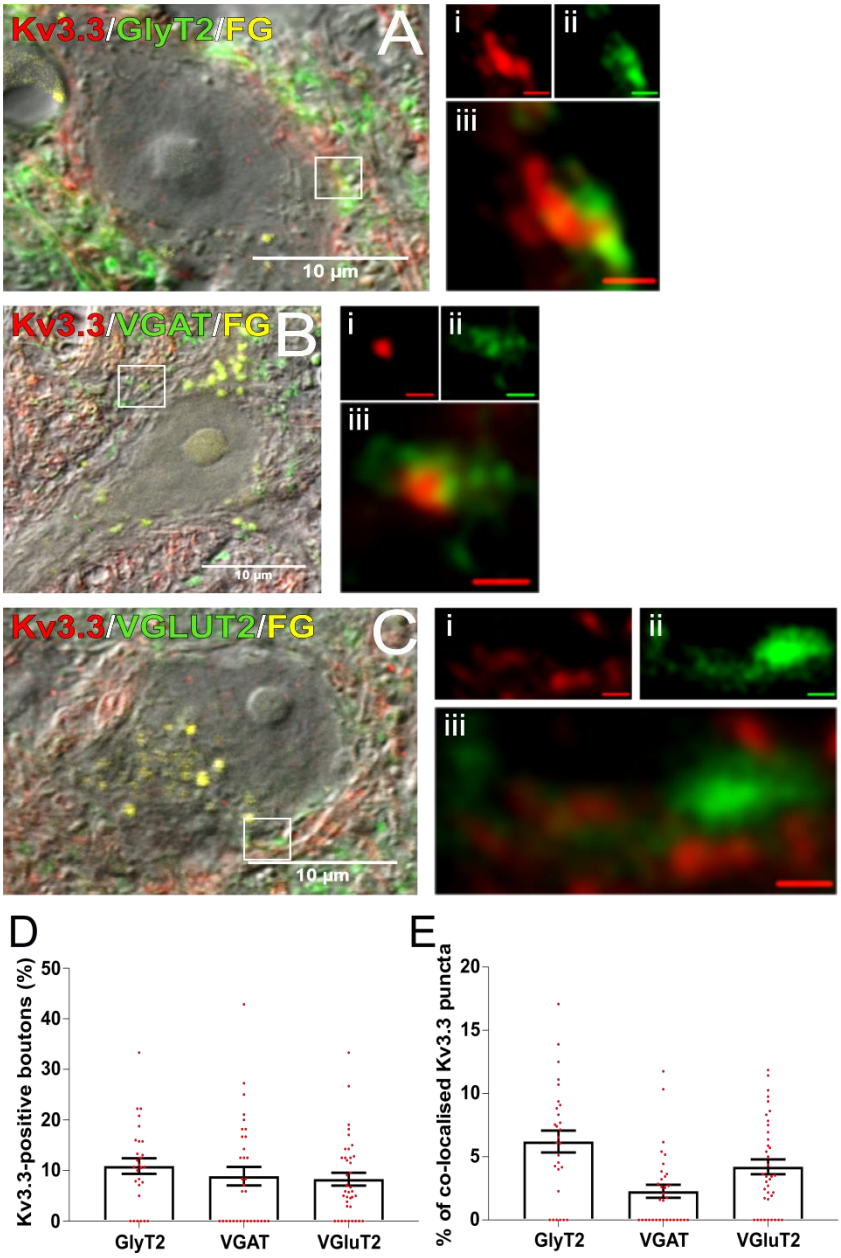


Figure 2.6. Kv3.3-IF puncta co-localised with inhibitory and excitatory synaptic markers onto PGN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.3 (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative PGN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord). Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii). Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

2.3.5 Kv3 co-localisation apposed to SPN motoneurones

Finally, for SPN motoneurones, 35% GlyT2, 19% VGaT and 15% VGluT2 boutons were co-localised with Kv3.1b puncta whilst 9%, 6% and 3% Kv3.1b puncta were co-localised with GlyT2, VGaT and VGluT2 boutons, respectively **(Fig. 2.7)**. For Kv3.3, 15% GlyT2, 22% VGaT, 12% VGluT2 boutons were co-localised with Kv3.3 puncta whilst 8%, 12% and 6% GlyT2, VGaT and VGluT2 were co-localised with Kv3.3 puncta (**Fig. 2.8)**.

In summary, Kv3.1 and Kv3.3 -IF co-localised with both excitatory and inhibitory synaptic markers in closely apposed to putative SPN motoneurones.

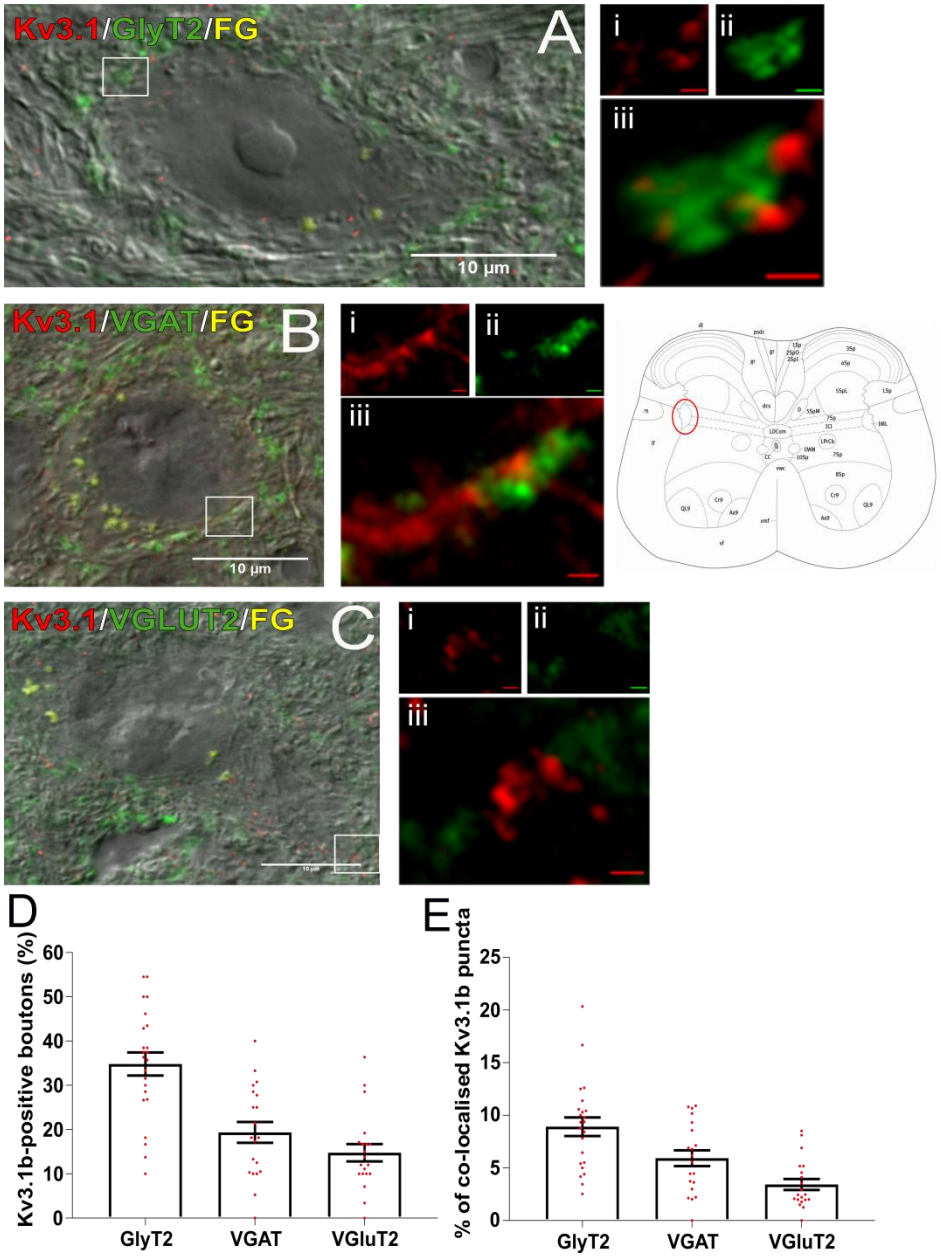


Figure 2.7. Kv3.1-IF puncta co-localised with inhibitory and excitatory synaptic markers onto SPN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.1b (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative SPN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord).* *Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii).* *Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

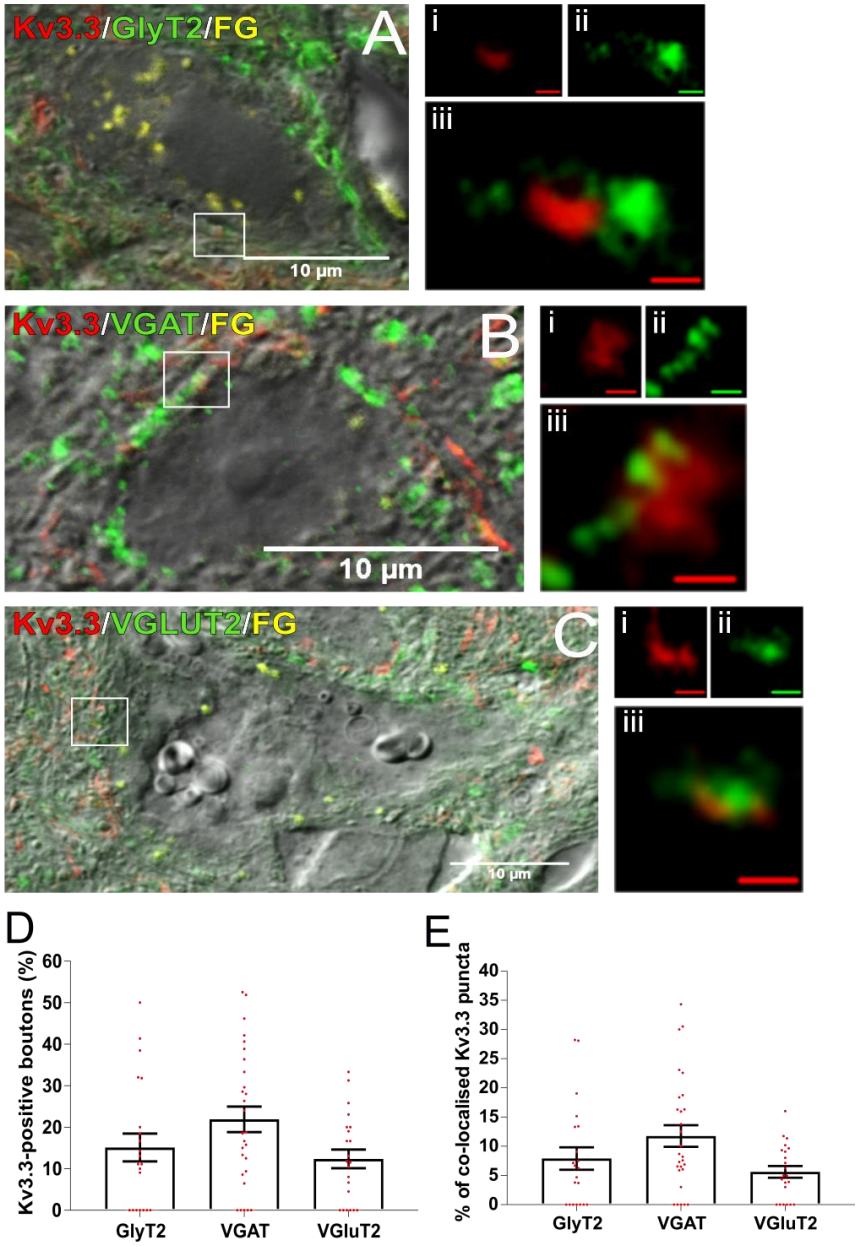


Figure 2.8. Kv3.3-IF puncta co-localised with inhibitory and excitatory synaptic markers onto SPN motoneurones.

A,B,C; *Triple labelling fluorescence immunohistochemistry of Kv3.3 (red), synaptic markers GlyT2 (A), VGAT (B), VGluT2 (C, all green), and Fluorogold-labelled (yellow) putative SPN motoneurones in the ventral horn of the lumbo-sacral spinal cord (circled in red in the diagram of the spinal cord).* *Superimposition with a DIC image reveals nuclei and the neuronal perimeter edged with synaptic bouton-like swellings. Airyscan images are also presented for a higher resolution visual inspection of the relationship between Kv3 puncta and synaptic markers (A,B,C ; i,ii,ii).* *Coloured scale bars indicate 0.5 µm.* ***D, E;*** *The mean co-localisation of synaptic boutons and puncta as a percentage of total boutons or puncta; each point represents data for an individual motoneurone and error bars are standard error of the mean (D, E). Data represents 3 neurones/section with 3 sections per mouse for 3 mice. Note some neurones had no co-localisation of synaptic markers with Kv3 puncta.*

In summary, both Kv3.1b and Kv3.3 puncta were co-localised with inhibitory and excitatory synaptic markers closely apposed to DLN, PGN and SPN motoneurones.

2.3.6 Are Kv3 channels functional in spinal synapses?

To examine whether Kv3 subunits were functional in spinal synapses, post-synaptic recordings were made from neurones in the lateral gray matter that showed excitatory post-synaptic potentials (eEPSPs) evoked in response to stimulation of descending tracts in the lateral white matter (Wang et al, 2010). Theoretically application of tetraethylammonium (TEA) should block pre-synaptic Kv3 channels and increase EPSP amplitude. In this paradigm cells were held at -100mV to avoid an effect due to post-synaptic TEA-sensitive Kv channels. Application of TEA (0.5mM) significantly increased the eEPSP amplitude of 4 cells out of 7 with 3 returning to baseline upon wash (Cell1 4.8 ± 0.7 mV to 8.5 ± 0.3 mV to 3.4 ± 0.3 mV, $p<0.001$; Cell2 5 ± 0.67 mV to 17 ± 0.97 mV to 4 ± 0.56 mV, $p<0.001$; Cell3 3 ± 0.48 mV to 7.3 ± 0.58 mV to 5 ± 0.59 mV, $p<0.001$; Cell 4 2.8 ±0.7 mV to 4.9 ± 0.7 mV to 4.3 ± 0.6 mV) **(Fig. 2.9 Aii).** The duration of eEPSPs from one of these four cells was significantly increased with application of TEA (Cell2, 165 ± 17 ms to 256 ± 17.7 ms to 118 ± 14 ms, $p<0.01$)**(Fig. 2.9Aiii).** One cell in which eEPSP amplitude was unaffected also displayed a significant and recoverable lengthening of eEPSP duration (Cell5, 260 ± 14.5 ms to 385 ± 43 ms to 273 ± 19 ms, $p<0.01$). The time taken to onset of an eEPSP was unaffected by application of TEA but showed clear stratification between cells **(Fig. 2.9 A iv)**, perhaps indicative of the distance from stimulation or the conduction velocity of the fibre.

These results suggested that some synapses were TEA sensitive and possibly contained Kv3 subunits, however, the effect of TEA on the time constant and steady state ohmic nature of the postsynaptic membrane cannot be separated from the presynaptic effect of TEA, thus the effect seen is potentially confounded.

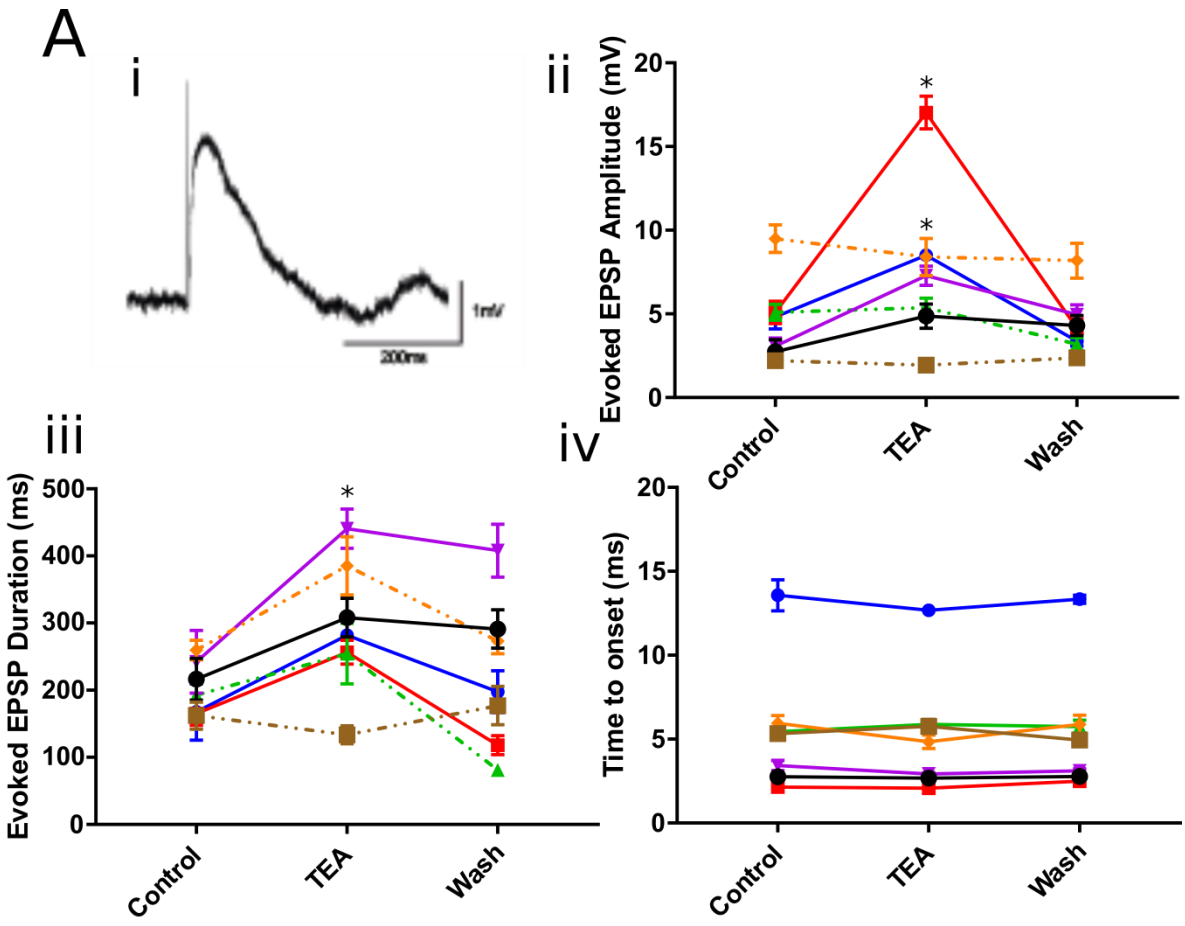


Figure 2.9. TEA increased the amplitude and duration of evoked EPSPs in some neurones

A; *The effect of 0.5 mM TEA application on evoked EPSP (eEPSP) amplitude and duration. An example of an evoked excitatory post-synaptic potential (eESPSP) (i). Measures of eEPSP amplitude (ii), duration (iii) and the time to onset (iv). Cells whose eEPSP amplitude statistically increased after administration of TEA are represented by solid lines in (ii) and (iii). Colour identifiers for each cell are consistent throughout. EPSPs were evoked with a 7 second interstimulus interval and at least 7 eEPSP per condition were analysed. P<0.05=*, p<0.01=**, p<0.001=***.*

2.3.7 Does the context of Kv3 immunoreactivity change with age?

Kv3 expression has been shown to decrease in other regions of the brain, therefore we established if the total and also excitatory/inhibitory context of Kv3 expression was susceptible to a change in age. The same immunohistochemical protocol was performed on tissue from 28 month mice, again looking at the DLN, PGN and SPN motoneurones of the lumbosacral spinal cord.

For the DLN motoneurones, there were no significant differences between 3 and 28 month animals in the number of VGluT2-IF boutons co-localised with Kv3.1b. However, there was a significant increase in co-localised VGAT-IF boutons co-localised with Kv3.3 immunoreactivity, from 10 ± 1.3 % to 22 ± 3.4 % ($p<0.01$) **(Fig. 2.10 A B, ii)**.

For PGN and SPN motoneurones, no significant changes were observed **(Fig. 2.10 C i).**

Thus, the context of Kv3 expression, the co-localisation with synaptic markers, changed very little between young and aged mice.

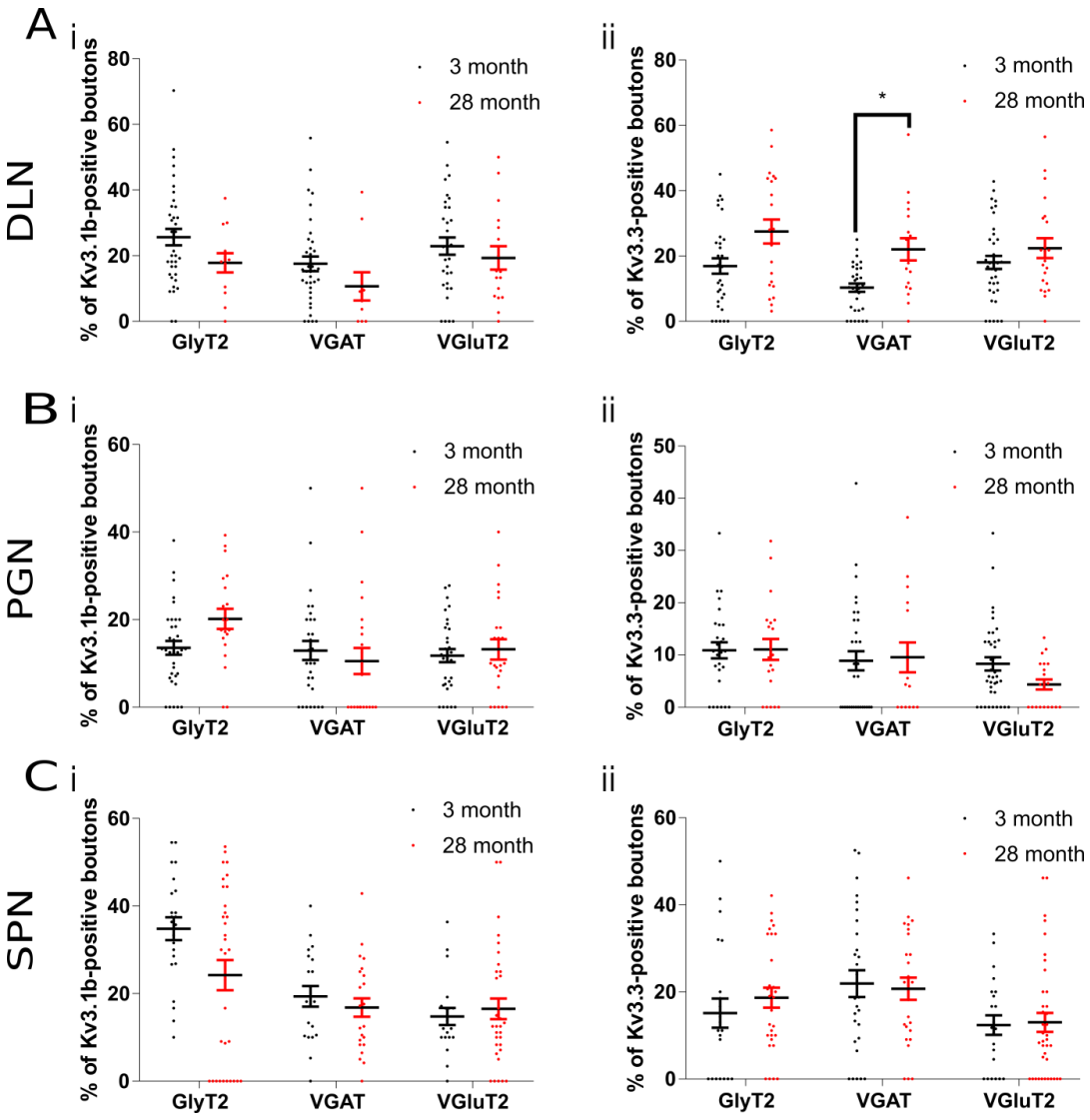


Figure 2.10. Kv3.3-VGAT co-localisation in DLN motoneurones changed with age.

A, B, C; *The number of normalised boutons, where immunoreactivity for synaptic markers co-localised with Kv3.1b (i) and Kv3.3 ii) immunoreactivity apposed to DLN (A), PGN (B) and SPN (C) motoneurones, in 3 month and 28 month mice. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***. Data represents 3 neurones/section with 3 sections per mouse for 3 mice.*

2.3.8 Does the expression of Kv3 channels change with age

The number of Kv3.1b puncta normalised to the perimeter of the cell membrane was relatively stable through age for DLN and PGN motoneurones (**Fig. 2.11 B, ii, iii**). There was however a significant reduction in Kv3.1b immunoreactivity for SPN motoneurones (125 ± 3.4 to 88 ± 3.9 puncta/100 μm, $p<0.001$) (**Fig. 2.11 B, iii**). Conversely, Kv3.3 was more susceptible to change in the vicinity of these neurones. Kv3.3 immunoreactivity was significantly reduced in DLN and PGN motoneurones (58 ± 1.68 to 49 ± 1.8 puncta/100 μm and 67 ± 1.5 to 44 ± 2 puncta/100 μm) **(Fig. 2.11 B, i, ii**). No change in Kv3.3 puncta apposed to SPN motoneurones was observed (**Fig. 2.11 B, iii**)

Importantly, these changes were observed in the context of a change in excito-inhibitory balance. In PGN neurones VGluT2-IF bouton expression was significantly reduced (33 ± 1.2 to 27 ± 1,3 boutons/ 100 μm) ($p<0.001$)(**Fig. 2.11 A, ii**). In SPN, VGluT2-IF boutons were also significantly reduced (26 ± 1.2 to 21 ± 0.9 boutons/ 100 μm (**Fig. 2.11 A, iii**). Synaptic bouton-IF in the DLN was stable between 3 month and 28 month (**Fig. 2.11 A, i**).

These results indicated a decrease in Kv3 and VGluT2 expression at bladder motoneurones in aged mice.

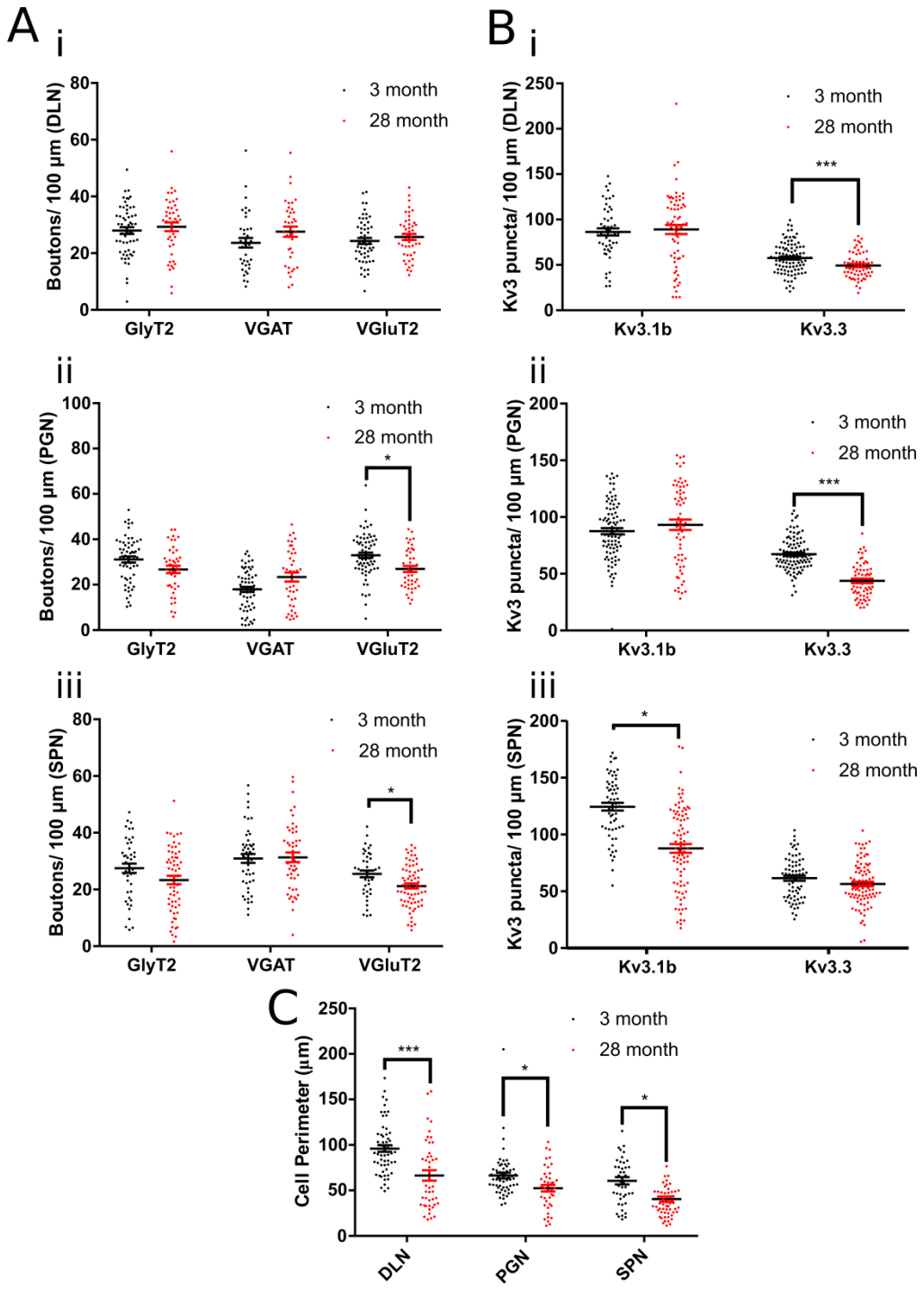


Figure 2.11. Kv3-IF and VGluT2-IF decreased with age

A; *The number of boutons immunoreactive for synaptic markers, GlyT2, VGaT and VGluT2, apposed to and normalised to the perimeters of DLN (i) PGN (ii) and SPN (iii) motoneurones in 3 month and 28 month mice.* ***B;*** *The number of Kv3 puncta apposed to and normalised to the perimeters of DLN (i) PGN (ii) and SPN (iii) motoneurones, is compared in 3 month and 28 month mice.* ***C****; cell perimeters for cells in the DLN, PGN and SPN at 3 and 28 month. . Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***. Data represents 3 neurones/section with 6 sections per mouse for 3 mice.*

2.4 Discussion

2.4.1 Kv3 immunoreactivity co-localised with synaptic markers

We found punctate Kv3.1b and Kv3.3 immunoreactivity co-localised with immunoreactivity of excitatory and inhibitory synaptic markers in close apposition to autonomic and somatic motoneurones postulated to be involved in the micturition reflex **(Fig. 2.3-2.8).** The type of immunoreactivity observed here concurs with that of other work in other areas of the spinal cord. Devaux et al. (2003) observed punctate Kv3.1b reactivity in the thoracic spinal cord and also in nodes of Ranvier and Brooke et al. (2006) observed Kv3.3 reactivity around the membranes of large cells in the ventral horn and co-localisation with presynaptic markers. Notably as previously described (Brooke et al., 2006), Kv3 immunoreactivity was absent from the intermediolateral horn. This concurrence, along with validation of these antibodies by other groups (Soares et al 2017), leads us to believe that the reactivity of the antibodies used in this study was specific. Speculatively, the presence of both subunits in the same types of synapses observed in this study suggests that Kv3.1b and Kv3.3 may exist as heteromers within these structure. Attempts to localise these synapses were made unsuccessfully using a proximity ligation assay (data not shown), where probes against antibodies for each subunit hybridise and fluoresce in situ if each subunit is within 50 nm of the other. Oher techniques such as co-immunoprecipitation would address this speculation.

In establishing the association of Kv3 channels with synaptic markers, it is important to explore the possible role and significance of these channels at this location. At the synapse, Kv3 channels are thought to constrain

neurotransmitter release as blockade of Kv3 channels increases calcium influx (Rowan et al., 2014), neurotransmitter release, decreases paired pulse ratios (Goldberg et al., 2005) and potentiates post-synaptic potentials (Ishikawa et al., 2003). It is therefore likely that Kv3 channels in this location play a similar role. Crucially, Kv3.3 channels have been reported to be fast inactivating, albeit not as fast as Kv3.4 channels. This type of inactivation would tune a synapse for activity dependent plasticity where repetitive stimulation inactivates the Kv3.3 channel, broadens the action potential to increase Ca2+ influx and neurotransmitter release. The role of Kv3.3 channels in this form of plasticity has so far been unexplored.

Perhaps one of the most important questions, is where are these glutamatergic, glycinergic and GABAergic synapses originating from? The answer is likely inhibitory and excitatory interneurones, several of which have been implicated in the micturition reflex and summarised by Shefchyk (2001). Briefly, local excitatory and inhibitory interneurones are found in close proximity to PGN motoneurones, with the latter eliciting IPSPs with a glycine and GABA component in addition to an unknown source of recurrent inhibition. Conversely, DLN motoneurones receive tonic excitation via a polysynaptic interneuronal pathway originating from segmental afferents during bladder filling. Inhibition of these DLN motoneurones for bladder voiding, involves a hyperpolarisation likely mediated by glycine receptors with a GABA component. A further potential source of GABAergic inhibition onto DLN motoneurones may arise from interneurones dorsal to the central canal that receive descending input from the PMC (Shefchyk, 2001). Clearly the sources of innervation to

putative motoneurones in the bladder reflex are diverse, but analysis of these interneurones and the functional synaptic connections they make to bladder motoneurones would be the first target in dissecting the role of Kv3 channels in synaptic transmission in the bladder reflex.

2.4.2 A fraction of synaptic boutons was Kv3-positive

In this analysis, a proportion of synaptic boutons and Kv3 puncta from 5% to 40% were co-localised with respect to the total immunoreactivity of each marker. This would indicate that the majority (>60%) of synaptic inputs onto bladder motoneurones were not associated with Kv3 channels. Similarly, the majority of Kv3 puncta was not associated with a single synaptic marker indicating that this immunoreactivity was associated with other structures such as passing axonal fibres and the post-synaptic membrane. The characterisation of Kv3 channels as both pre and post-synaptic in the spinal cord has previously been highlighted using electron microscopy by Brooke *et al.*, (2006). However, Kv3 immunoreactivity was found to be not in the post-synaptic membrane of ventral motoneurones at the ultra-structural level. Furthermore, some neurones analysed had no co-localisation of Kv3 puncta and inhibitory and excitatory synaptic markers, however, VGluT2 is not a universal marker for all excitatory synapses and we cannot rule out co-localisation with VGluT1 and VGluT3 immunoreactive structures. This observation was more pronounced around SPN and PGN autonomic motoneurones and may reflect subsets of neurones in this pool that do not receive synaptic inputs that are Kv3-positive, suggesting that transmission to autonomic motoneurones is influenced by Kv3 channels to a lesser degree than somatic motoneurones.

2.4.3 EPSPs exhibited TEA sensitivity suggesting functional involvement of Kv3 channels in synaptic transmission

EPSPs evoked by stimulation of descending fibres in the lateral gray matter were in some cases TEA sensitive, meaning EPSP amplitude was increased. In these experiments the hyperpolarised holding potential (-100 mV) eliminated any effect being due to TEA sensitive Kv channels on the post-synaptic membrane. Kv3 channels are typically associated with GABAergic neurones and structures, with some exceptions (Dallas *et al.*, 2005; Alle, Kubota and Geiger, 2011). The association of Kv3 channels in excitatory structures in the lumbosacral spinal cord is a relatively novel idea, therefore we focussed on evoked excitatory postsynaptic potentials. However, an important consideration is that TEA will block presynaptic Kv1 channels as well as pre synaptic Kv3 channels (Hoppa *et al.*, 2014), however in the Calyx of Held, the pre-synaptic role of Kv1 channels appears to be to reduce aberrant excitability rather than reduce neurotransmitter release such as by Kv3 channels (Ishikawa *et al.*, 2003). Thus while the action of TEA is unselective between Kv1 and Kv3 channels the increased amplitude is likely underlined by Kv3-mediated increased neurotransmitter release.

2.4.4 Age-related susceptibility of spinal excitatory-inhibitory balance

We observed changes in excitatory synaptic markers with age. In contrast with unpublished data (Merican, 2016) we observed no changes in synaptic markers in the DLN. We did however see a decrease in markers of VGluT2 in SPN motoneurones in concurrence with Merican (2016). This decrease in glutamatergic synapses also agrees with the work of Santer *et al.*, (2002) who observed a reduction in the area of glutamatergic synaptic contact at the ultrastructural level around sympathetic preganglionic motoneurones. By what mechanism are excitatory synapses reduced and what are the implications for a decrease in excitatory synapses in these circuits? A relatively recent study into the classical complement cascade suggests that key mediators of the cascade that eliminate synapses are up-regulated in the CNS during ageing (Stephan et al., 2013). In this cascade, the protein C1q targets structures designated for phagocytosis by macrophages, was found close to synapses and elevated by 300 fold during ageing. Perhaps these mediators are also upregulated around the synapses onto bladder motoneurones and are responsible for the glutamatergic losses seen here. Clearly a reduction in the glutamatergic innervation of autonomic motoneurones involved in the micturition would mean that these neurones are likely to be less active and perhaps this represents a central mechanism by which the function of the bladder is dysregulated in ageing.

2.4.5 Kv3 immunoreactivity changed with age

In the context of these synaptic changes, Kv3 subunits were also changed. In the DLN and PGN, Kv3.3 immunoreactivity was decreased and in the SPN, Kv3.1b immunoreactivity was decreased. However, despite the decreased

immunoreactivity of Kv3 subunits, co-localisation was relatively stable with age. Many possibilities could explain this phenomenon. Firstly, loss of Kv3 subunit expression could be occurring ubiquitously across excitatory and inhibitory synapses such that large changes are statistically imperceptible. Secondly, other types of synapses exist that haven't been investigated here, such as GAD65 axo-axonal synapses that terminate onto VGluT1 synapses for primary afferent depolarisation (Mendel et al, 2016). In addition and finally, many Kv3 puncta weren't attributable to the presynaptic markers used here, and perhaps a decrease in Kv3 expression is occurring in other structures, synaptic or not, closely apposed to the motoneuronal membrane. Despite this stability, what was especially interesting was that co-localisation of Kv3.3 in the DLN with inhibitory marker VGaT was significantly increased, even in a context of an overall reduction in Kv3.3 immunoreactivity. However, while there is some evidence that ion channel mRNA and protein levels remain relatively stable with age (up to 20 months) in many other CNS regions (Boda et al., 2012), the present findings indicate that this is not the case at advanced age (28 months). Here we have demonstrated the importance of the changes in the context of Kv3 expression that may have physiological implications in ageing.

Chapter 3

3 Novel modulation of Kv3 heteromers and spinal interneurones

3.1 Introduction

3.1.1 Kv3 biophysics

Kv3 channels are tetrameric pore-forming structures in the cell membrane that facilitate the movement of K+ ions across an otherwise impermeable lipid membrane. They are voltage-activated, meaning that the channel structure is coupled to changes in the distribution of charge and thus the electrical potential (voltage) across the membrane. This results in structural states where the channel is either open and conducting or closed and non-conducting or in transition between. The terms activation, deactivation and inactivation refer to the channel opening, the channel closing and the channel entering a state in which it can no longer conduct properly. To study these channels, an experimenter can control the electrical potential across a cell and record when activation, deactivation and inactivation occur and how long each process takes.

While the opening and closing of Kv3 channels is determined by the electrical potential, the actual motivation for K+ ions to move from inside the cell (intracellular) to outside the cell (extracellular) is determined by both an electrical and chemical gradient acting on the ion. Essentially, like for like charges repel and particles in high concentration prefer to spread out. At rest the chemical gradient is established by an ATP driven transporter (Na+K+ATPase) that favours intracellular K+ and extracellular Na+ accumulation. Therefore, when Kv channels open, K+ ions flow out of the cell

until a balance between the electrical and chemical gradient is achieved. The relationship between the equilibrium potential (Ek) and the electrochemical gradient is given by the Nernst Equation:

Equation 3.1

$$Ek = \frac{RT}{zF} \ln \frac{[Ko]}{[Ki]}$$

where R is the gas constant, T is temperature in Kelvin, z is the valency of the ion, F is Faraday's constant and [Ko] and [Ki] are the extracellular and intracellular concentrations of potassium.

There are four types of Kv3 subunits that can tetramerise (Kv3.1-4) to form either homomers (a channel of the same subunits) or heteromers (a channel of different subunits) and there is a good deal of literature on the biophysics of these channels as homomers in several types of expression systems (see General Introduction).

Structurally, there are some important differences between the subunits that determine whether the homomeric channel is a steadily conducting delayed rectifier like Kv3.1 and Kv3.2 or a fast inactivating A-type current like Kv3.3 and Kv3.4. Typically, the rapid inactivation of Kv3.3 and Kv3.4 channels is due to a "ball and chain" structure at the intracellular N-terminus that acts to plug the pore upon activation, although there are other modifications, such as

phosphorylation, that these channels can undergo that affect the rate inactivation (see General Introduction).

In general, these channels activate rapidly at potentials likely only achieved during action potentials and deactivate as quickly as the membrane repolarises back to negative potentials. This specialises them for efficient repolarisation of action potentials and recovery of Na+ channels from inactivation and for keeping refractory periods short. Kv3 subunit expression often overlaps and it is likely that in a physiological setting in native neurones they tetramerise to form heteromers with distinct kinetics.

3.1.2 Kv3 pharmacology

Kv3 channels are pharmacologically blocked by internal and external application of TEA and by 4-AP, BDS toxins and gambierol (see General Introduction). Relatively recently a series of imidazolidinedione derivatives that are small lipophillic and cell permeant molecules such as AUT1 ((5R)-5-ethyl-3-(6-{[4-methyl-3-(methyloxy)phenyl]oxy}-3-pyridinyl)-2,4-imidazolidinedione) have been shown to potentiate Kv3.1, Kv3.2 and Kv3.3 currents at specific voltage steps by increasing the open probability of the channel (Rosato-Siri *et al.*, 2015; Taskin *et al.*, 2015; Brown *et al.*, 2016; Boddum *et al.*, 2017). However, detailed study on the effect of these compounds on Kv3.4 channels and on heteromers that may relate to endogenous Kv3 channels in the spinal cord is lacking. Therefore it is important to understand the effect of these new compounds on all Kv3 subunits and channels that may be reflect endogenous heteromers in native neuronal tissue, in order to interpret phenotypic changes during administration of the compound.

3.1.3 Kv3 heteromers

Whilst Kv3 subunits are often co-expressed and sometime co-precipitated in the same neurones and the likelihood of them forming functional heteromers is high, little biophysical evidence in native neurones exists to demonstrate the diversity of heteromer kinetics and their functional relevance (Hernández-Pineda *et al.*, 1999; Baranauskas *et al.*, 2003. However, there are two examples of heteromeric Kv3 channels in recombinant cell lines, a Kv3.1b/Kv3.2a (Lewis, McCrossan and Abbott, 2004) and a Kv3.1b/Kv3.4a heteromer (Baranauskas *et al.*, 2003). In these examples, heteromer kinetics only appear to be distinct when the individual kinetics are distinct from each other. For example, Kv3.1b and Kv3.2a are biophysically similar and the resulting current with co-expression of the two channels in CHO cells was indistinguishable from the Kv3.1b and Kv3.2a homomeric currents (Lewis, McCrossan and Abbott, 2004). However, Kv3.1b and Kv3.4a co-expressed in HEK cells, one a delayed rectifier and the other an A-type channel, produced a unique channel with differences in the rate of inactivation and the voltage at which conductance was half the maximum (V50) (Baranauskas *et al.*, 2003). In light of this, the importance of coassembly of biophysically similar subunits probably lies in how other proteins interact and modulate the heteromer, whereas for distinct subunits, heteromerisation likely provides a novel channel with a specific purpose for the cell it is formed in.

In the spinal cord, the presence of Kv3.1, Kv3.3, and Kv3.4 mRNA, protein and immunoreactivity has been confirmed (Perney *et al.*, 1992; Weiser *et al.*, 1994; Deuchars *et al.*, 2001; Brooke *et al.*, 2002, 2004). Curiously, Kv3.4 expression

levels are relatively low in the spinal cord compared to other subunits and may act as heteromeric partners for Kv3.1 and Kv3.3 subunits that are highly expressed in the spinal cord. As discussed in General Introduction, Kv3.1b positive interneurones are antecedent to autonomic motoneurones in the lateral grey matter. Futhermore, results from Chapter 1 indicated that autonomic motoneurones in the lateral aspect of L6-S1 receive glutamatergic, glycinergic and GABAergic innervation and in the lumbosacral spinal circuitry of the micturition reflex, this GABAergic input is likely to originate from interneurones in the local vicinity of the motoneurones pool. By patch clamping neurones in this region, it is possible to explore the effect of AUT1 application on interneuronal firing and extrapolate as to how that may modulate autonomic outflow in the spinal cord. Kv3 channels in other CNS regions often endow neurones with fast firing properties and the application of AUT1 to these neurones has produced a mix of reported effects, from suppressing excitability to increasing firing frequency to having no effect at all (Brown et al, 2016, Taskin et al, 2015, Boddum et al, 2017, Olsen et al, 2018). The effect of this compound on Kv3-expressing interneurones of the spinal cord is unknown and investigating this neuronal population in the vicinity of autonomic bladder motoneurones would provide an avenue to consolidate these mixed reports and characterise interneurones at the level of bladder circuitry.

3.1.4 Aims

- **To understand the effect of AUT1 on Kv3.4a homomers and Kv3.1b/Kv3.4a heteromers using recombinant systems to study each channel in isolation.**
- **To understand the effect of AUT1 on the electrophysiology of native lumbosacral spinal interneurones.**

3.2 Materials and Methods

3.2.1 Cell culture of HEK293T cells

HEK cells were cultured in modified Eagles Medium (MEM, Sigma-Aldrich) with fetal bovine serum (qualified E.U.-approved, South America origin, Gibco), L-glutamine, penicillin and streptomycin (PenStrep, 5000 U/ml, Cat. no. 15070063, Gibco) in sterile flasks (T25 and T75) in an incubator at 37 °C 5% CO_2 , 95% O_2. When cells reached 80% confluency they were washed in sterile 1x Phosphate Buffered Saline (PBS), trypsinised (Trypsin-EDTA (0.05%, phenol red, Cat. No 25300054), Gibco) and removed to a falcon to be centrifuged for 10 minutes at 2000 rpm. The resultant pellet was resuspended in fresh MEM media and a fraction of this suspension was seeded into fresh flasks or 6 well plates for storage or experiments, respectively.

Human embryonic kidney (HEK) cells are a reliable cell line amenable to transfection of human cDNA and are commonly used to express and study Kv channels. Endogenously they express a number of Kv channels; mRNA for delayed rectifiers Kv1.1, Kv1.2, Kv1.3, Kv1.6 and Kv3.1, as well as mRNA for

transient IA types Kv1.4, Kv3.3, Kv3.4 and Kv4.1 have been detected, with a total outward current of about 200pA (Jiang *et al.*, 2002). Therefore it is crucial that over expressed Kv currents are significantly larger than these endogenous channels.

3.2.2 Transfection of HEK293T cells with Kv3.1b and Kv3.4a cDNA

400,000 HEK 293T cells were seeded into the wells of 6 well plates. Following 24 hours of expansion the wells were transiently transfected with human Kv3.1b, Kv3.4a and GFP cDNA; 0.5ug/ul was used for homomeric experiments and a 2:1 ratio of Kv3.1b to Kv3.4 a, 1ug/ul to 0.5ug/ul, was used for heteromeric experiments. Transfections followed the Dreamfect Gold™ (Oz Biosciences) protocol with the volume of DreamFect lipofectamine kept within a 2:1 ratio of total cDNA mass. In brief, cDNA and lipofectamine were allowed to complex in eppendorfs for 20 minutes at room temperature before addition to the well MEM media. Cells were incubated in the DNA/DreamFect media for 12 hours before replacement of the transfection media with fresh MEM media. Due to large currents, experiments were performed on transfected cells expressing GFP >100 hours post transfection.

3.2.3 Voltage clamp recordings of transfected HEK293T cells

Transiently transfected HEK cells were seeded onto circular glass coverslips (200,000/coverslip) and adhered overnight (16 hours) at 37 ^{o}C . For voltage clamp recordings, coverslips were placed in a recording chamber underneath an upright Leica microscope equipped with a camera (COHU 4912-3010/ER3791A). The chamber was continuously perfused with an extracellular solution composed of; NaCl, 137 mM ; KCl, 4 mM; glucose, 10 mM ; HEPES, 10

mM; CaCl2, 1.8 mM, and MgCl2, 1 mM, osm=308, pH7.3 with NaOH (5M). Thick-walled borosilicate glass pipettes of 1.8-3.5 MΩ were fabricated using a vertical puller (Narishige, PC-100) and filled with an intracellular solution composed of; KCl, 120 mM; EGTA, 10 mM; KOH, 31.25; Mg2ATP, 4 mM; HEPES, 10 mM; CaCl2, 5.37; MgCl2, 1.75, pH7.2 with KOH. A 4 mV LJP was not corrected for. Cell membrane potentials were held at -70mV and series resistance was compensated by 70% (Kv3.1b) or 80 % (Kv3.4a). Recordings were made at room temperature using a Multiclamp 700A, acquired at 50 kHz, filtered at 8 kHz using a 4 pole Bessel filter, digitised using a Digidata 1440A and recorded in Clampex 10 software.

For all types of protocol cells were held at -70 mV.

For activation experiments, the voltage of the cell was incremented in 10 mV 200 ms steps from -90 mV to 50 mV. For cells expressing Kv3.4a the duration between each step was 4 s to allow for recovery of inactivated channels, whereas 1 s for Kv3.1b HEK cells sufficed.

Deactivation experiments for Kv3.1b involved a 20 ms pulse to 50 mV to activate channels, followed by steps to potentials between -90 mV and 0 mV. Tail currents were fitted to obtain the time constant (τ or tau) for the deactivating current as a function of voltage. Tail currents, however, for Kv3.4a channels were fitted at -70 mV after a 12 ms step to 40 mV during inactivation experiments (see below). Within this timeframe Kv3.4a channels hadn't fully inactivated and so deactivating currents were observed.

Inactivation protocols consisted of a 20s (Kv3.1b) or 5s (Kv3.4a) pre-step incremented in 20mV steps from -70 mV to 30 mV. Each pre-step was followed by a 200 ms (Kv3.1b) or 12 ms (Kv3.4a) test step to 40 mV and a 50s recovery at -70mV between steps.

Axon Text Files were created in Stimfit from waveforms extracted from spinal cord neurones in Axon Binary File recordings. Axon Text Files can be incorporated into Clampex protocols and used as a voltage command waveform to simulate membrane changes during an action potential.

Control (0.1% DMSO) measurements were obtained 3-4 minutes after break in to allow for perfusion of the intracellular solution. AUT1 (10 μM, Autifony, Therapeutics Ltd) in 0.1% DMSO was perfused in extracellular solution and measurements were obtained at least 3 minutes after application of the compound. Wash off experiments were conducted in two cells where access resistance and holding current were stable for the duration of the experiment.

3.2.4 Kv3 current analysis

Cells were excluded from analysis if channel currents were too large, holding currents exceeded -300 pA, or access resistance changed by > 10%.

Current traces were analysed in Stimfit using custom python scripts (see Appendices). In brief, linear pipette leak current was calculated by plotting a straight line through 0 mV/pA and the current during a -90 mV step, close to Nernst potential of -93 mV where there is no net current of K+ **(Fig. 3.1)**. Peaks, plateaus and tail currents were measured and fit to obtain activation, inactivation and deactivation tau **(Fig. 3.2)**. The voltage error over

uncompensated series resistance was calculated, currents corrected and used to calculate the conductance using:

Equation 3.2

$$I = g(E - Ek)$$

Or

Equation 3.3

$$I = gK(V,t)V\left(\frac{\exp((qV - Ek)/kT) - 1}{\exp((qV)/kT) - 1}\right)$$

where I is current, gK is conductance, V is voltage, k is Boltzmann's constant, T is temperature in Kelvin, Ek is the Nernst potential for K+ and q is the charge of an electron.

Conductances were normalised to the maximum to produce G/Gmax plots which were fit with a Boltzmann sigmoidal **(Fig. 3.3, 3.4)** to estimate the voltage at which 50% of the channels were conducting (V50):

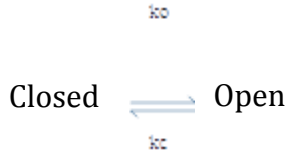
Equation 3.4

$$Y = Bottom + \frac{Gmax - Bottom}{1 + \exp\frac{V50 - V}{Slope}}$$

where Gmax is the maximal conductance, V50 is voltage at half-maximum conductance, V is voltage and slope represents the voltage dependence of the conductance.

Currents evoked by action potential and train waveforms recorded from spinal neurones were fit, peak amplitude was obtained and latency from action potential onset and voltage peak to current peak was measured.

A simplified Markov model describing the conductance and activation kinetics was created using the following scheme:



where ko and kc were described by:

Equation 3.5

$$ko = ko(0)\exp\left(\frac{zoFV}{RT}\right)$$

Equation 3.6

$$kc = kc(0)\exp\left(\frac{-zcFV}{RT}\right)$$

where ko(0) and kc(0) are ko and kc at 0 mV, zo and zc reflect the voltage dependence of ko and kc, F is Faradays constant, R is the gas constant and T is temperature in Kelvin.

Values for Ko, Kc, zo, zc were obtain by simultaneously fitting conductance data to:

Equation 3.7

$$Popen = ko / (ko + kc)$$

with activation tau (τ) to:

Equation 3.8

$$\tau = 1/(ko + kc)$$

using the least squares algorithm of the lmfit module in python.

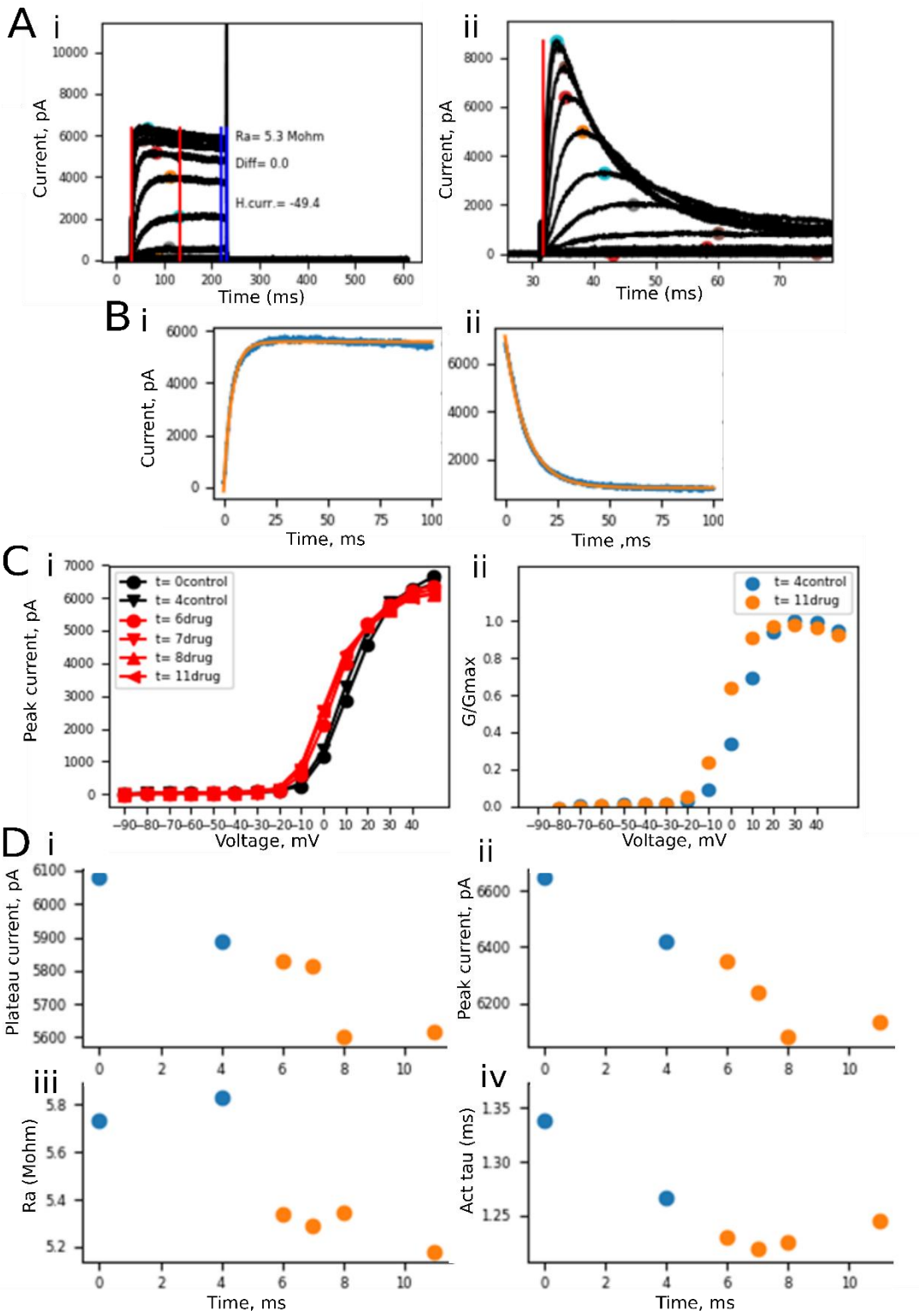

A i
Current, pA
Time (ms)
Ra= 5.3 Mohm
Diff= 0.0
H.curr.= -49.4
ii
Current, pA
Time (ms)
B i
Current, pA
Time, ms
ii
Time ,ms
C i
Peak current, pA
t= 0control
t= 4control
t= 6drug
t= 7drug
t= 8drug
t= 11drug
Voltage, mV
ii
G/Gmax
t= 4control
t= 11drug
Voltage, mV
D i
Plateau current, pA
ii
Peak current, pA
iii
Ra (Mohm)
Time, ms
iv
Act tau (ms)
Time, ms

Figure 3.1. Voltage clamp analysis.

A) Peak currents were measured between the red lines and steady state currents between the blue lines for Kv3.1b (i) and Kv3.4a (ii) channels. Holding current was reported to evaluate stability between time points. Each current file was associated with an access resistance file (Ra) and the Ra and the percentage difference from the first time point was visualised. B) Single exponential fits for the activation and inactivation (where appropriate) were used to measure the activation (i) and inactivation (ii) tau. C) Conductances were calculated based on the current and electrochemical driving force and fit to Boltzmann functions. D)Current rundown (I, ii), access resistance (iii) and activation/inactivation kinetics (iv) were monitored over time. The effect of the AUT1 compound was visualised over time. Orange markers indicate in the presence of compound and blue markers indicate control.

3.2.5 Statistics

Paired t-tests were conducted in Graphpad Prism software. To avoid conducting multiple comparisons and inducing statistical error, fits, rather than individual values, for G/V plots, τ/V plot, AP waveform plots were statistically compared in Graphpad Prism using the extra sum-of-squares F test. Significance here is presented by asterisks, where one fit cannot describe both datasets being compared.

3.2.6 Current clamp of TEA sensitive spinal neurones

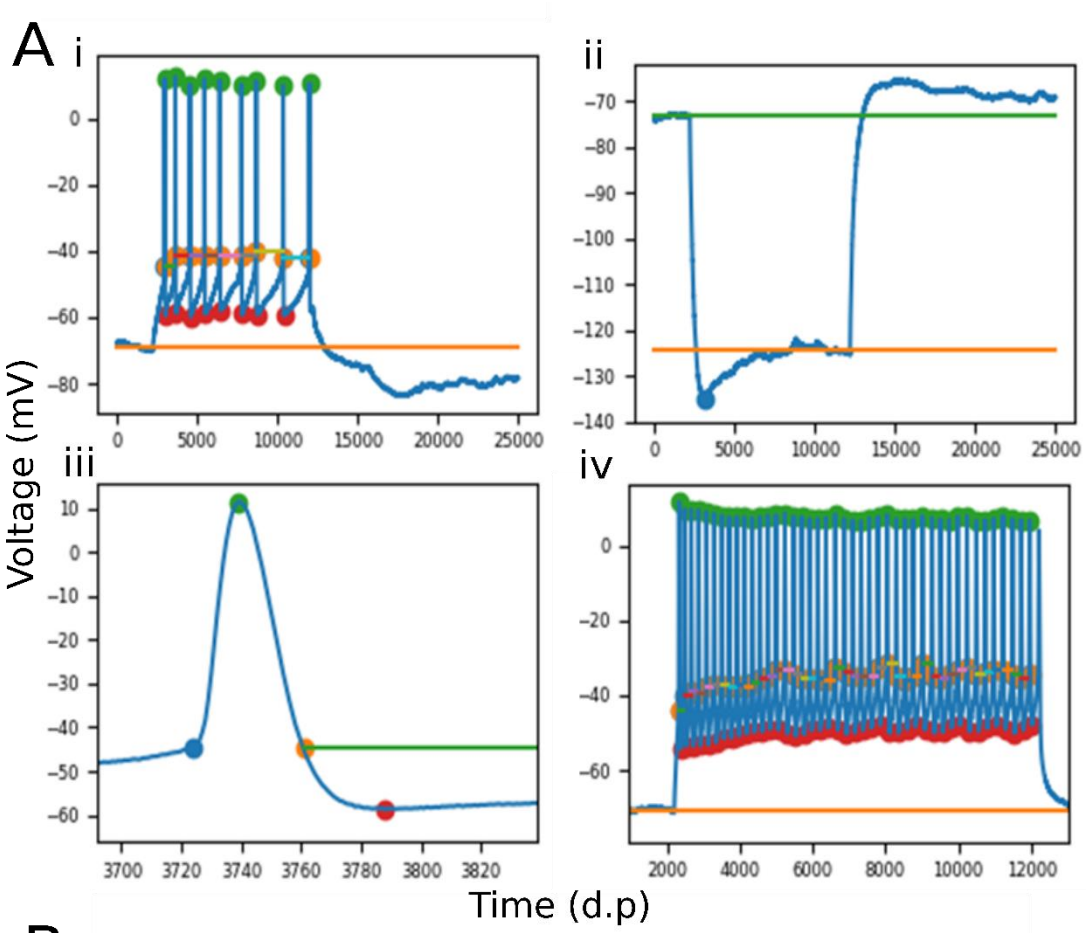
Tissue preparation and recordings were carried out at as described in Chapter 1. Spinal cord slices were pre-incubated in either control aCSF or 10 μM AUT1 based on advice from Autifony regarding the time to onset of action in slice electrophysiology. The resting membrane potential was recorded and negative current was injected to hold the cell at -70 mV. Cells in which the holding current exceeded -250 pA were excluded. Cells were characterised by depolarising and hyperpolarising 1 second (1s) current steps from -130 to 280 pA and their sensitivity to TEA (0.5 mM).

3.2.7 Analysis

Recordings were analysed in Stimfit (Guzman et al, 2014) using custom python scripts In brief, a user-defined dV/dt threshold was used to automatically segment the start and end and after-hyperpolarisations of all action potentials in a sweep **(Fig. 3.5, 3.6, 3.7)**. Between these points an array of measurements were made **(Table 3.1, Fig. 3.8)**:

Cell features	**AP features**	**AHP features**	**Firing Features**
Membrane time constant	Threshold	AHP duration	Number of AP
Membrane resistance (calculated by a fitting straight line through voltage responses to hyperpolarising current steps)	Latency to first AP	AHP amplitude	Instantaneous Freq.
Cell capacitance (estimated from membrane tau)	AP amplitude	AHP max decay	Steady State Freq.
	AP duration		Interspike Interval Coefficient of Variation
	AP repolarisation duration		
	AP max decay		
	AP rise/decay Ratio		

Table 3.1. Current clamp features analysed in Stimfit


A
i
ii
iii
iv
Voltage (mV)
Time (d.p)

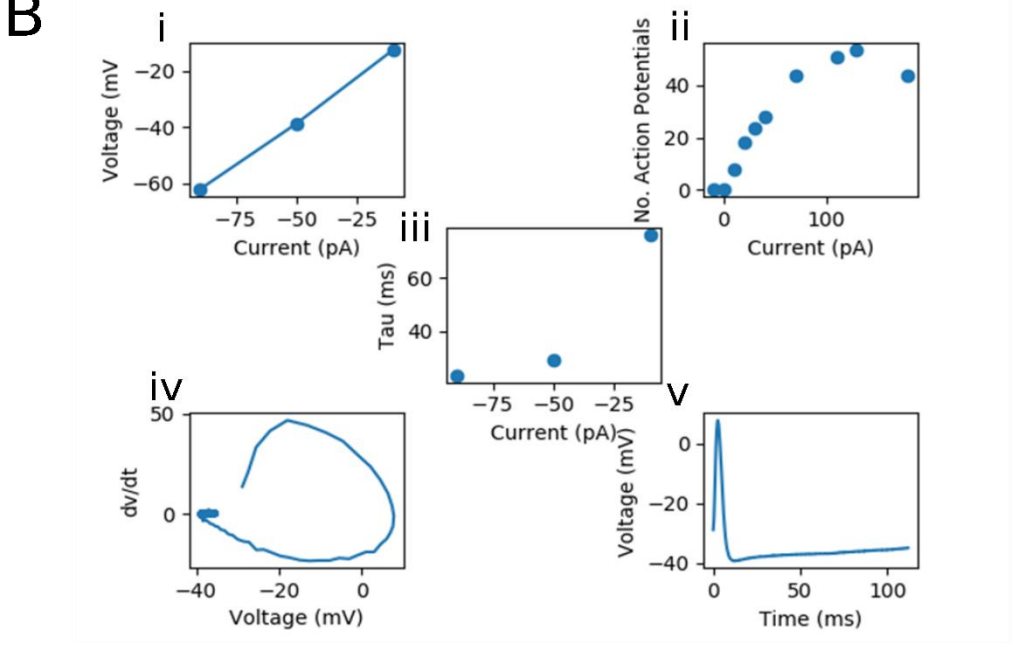

B
i
Voltage (mV
Current (pA)
ii
No. Action Potentials
Current (pA)
iii
Tau (ms)
Current (pA)
iv
dv/dt
Voltage (mV)
v
Voltage (mV)
Time (ms)

Figure 3.2. Current clamp analysis

A)Blue and orange dots mark the start and end of each action potential, green dots mark the action potential peak and red dots mark the AHP minima. AHP duration is marked by a line from the AP end to where membrane voltage approaches the voltage at the AP end (i). The presence of Ih was calculated by subtracting the minima from the steady-state voltage response (ii). B) Steady state voltage response to hyperpolarising current injections and the gradient from a linear fit derived the membrane resistance (Rm) (i). Frequency of firing against current injection (ii). Membrane time constant (tau) measured fitting a single exponential to hyperpolarising changes in membrane voltage (iii). Phase plot of average first action potential at rheobase (iv). Average first action potential at rheobase (v).

3.3 Results

3.3.1 Comparison of Ohmic normalisation vs GHK normalisation

Ohms law is often used to derive the conductance of an ion channel. However, this law describes a linear relationship between the electrical driving force and current evoked where a non-linear relationship actually exists. Therefore, conductance is more accurately derived by GHK normalisation of the electrical driving force described by Equation 3.3 (Clay, 2000). It is unclear whether this normalisation has been used to describe the conductance of Kv3 channels before, however its use here described channels with a steeper voltage dependence, shifted V50 and earlier peak. For HEK cells transfected with Kv3.1b cDNA, peak conductance was achieved 10 mV earlier due to a steeper voltage dependence using GHK normalisation **(Fig. 3.9A).** The effect on HEK cells transfected with Kv3.4a alone and co-transfected with Kv3.1b andKv3.4a was more pronounced, describing a conductance that peaked 20 mV earlier and with a more negative V50 **(Fig. 3.9 B, C)**. All conductance data presented herein uses GHK normalisation instead of ohmic normalisation.

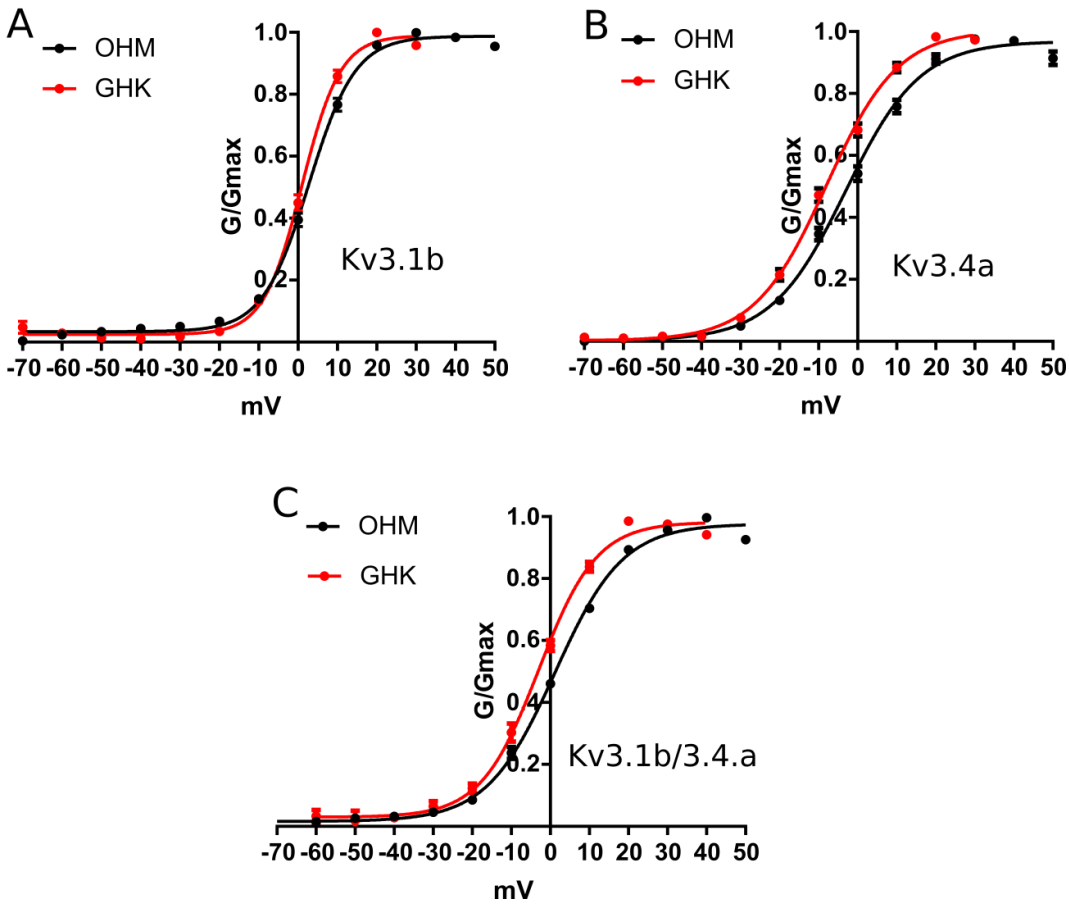


Figure 3.3. GHK normalisation steepens and negatively shifts voltage dependence

A,B,C; *Conductances normalised using Ohms law and GHK equation (see Materials and Methods) for HEK cells transfected with Kv3.1b cDNA, Kv3.4a cDNA and both Kv3.1b/3.4a cDNA. Error bars are SEM.*

3.3.2 Kv3.1b HEK transfection produced delayed rectifer currents

Cells transfected with Kv3.1b cDNA exhibited a delayed rectifier current **(Fig. 3.10 Ai),** with slow C-type inactivation **(Fig. 3.10 Aii)** and rapid deactivation back to resting potential **(Fig. 3.10 Aiii).** The channel began to activate between -20 and -10mV with a steep voltage dependence and peak conductance at 20mV **(Fig. 3.10 B).** Long depolarising pre-step pulses showed Kv3.1b began to inactivate at -30mV with full inactivation occurring at 10mV **(Fig. 3.10 B).** Activation tau of Kv3.1b from -70 to 50 mV was 1.2 ± 0.05 and deactivation tau of Kv3.1b channels upon repolarisation to -70 mV was 0.22 ± 0.08 ms **(Fig. 3.10 C).**

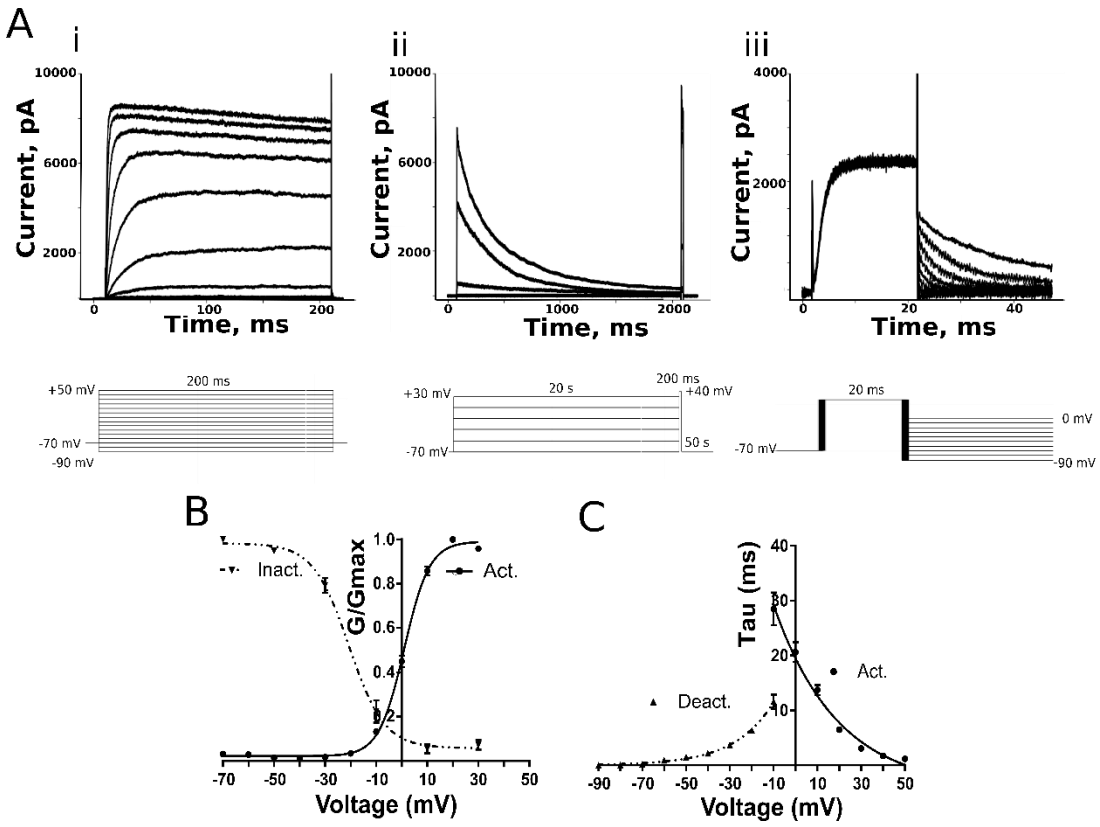


Figure 3.4. HEK transfection with Kv3.1b cDNA produced high-voltage activating delayed rectifier currents

A) *Representative Kv3.1b current traces for activation (i), inactivation (ii) and deactivation (iii)* ***B)*** *Overlay of normalised Kv3.1b conductances during inactivation (dashed line) and activation (solid line) protocols fit to Boltzmann functions.* ***C)*** *Activation (solid line) and deactivation (dashed line) tau for Kv3.1b measured from activating and deactivating currents and fit to mono-exponential functions. Error bars are SEM.*

3.3.3 AUT1 treatment negatively shifted activation of Kv3.1b currents and slowed rate of deactivation

In an example of a 0 mV step, Kv3.1b currents were evidently potentiated by AUT1 **(Fig. 3.11 A).** Treatment with 10 μM AUT1 shifted the V50 of activation by 8.6 mV (0.21 ± 0.82 mV to -8.4 ± 1.7 mV, n=5) and shifted the V50 of inactivation by 8.8 mV (-22 ± 1.7 mV to -30.8 ± 2.3 mV, n=4) to more negative potentials **(Fig. 3.11 C, Ei, Fi).** Neither slope nor Gmax were affected in each protocol **(Fig. 3.11 E, F ii, iii).** A single exponential fit could describe the activation tau of Kv3.1b in both control and AUT1 indicating no significant changes of the compound on the rate of activation. However, deactivation kinetics were significantly slowed with treatment of AUT1, nearly 4 fold at -70 mV **(Fig. 3.11 D).**

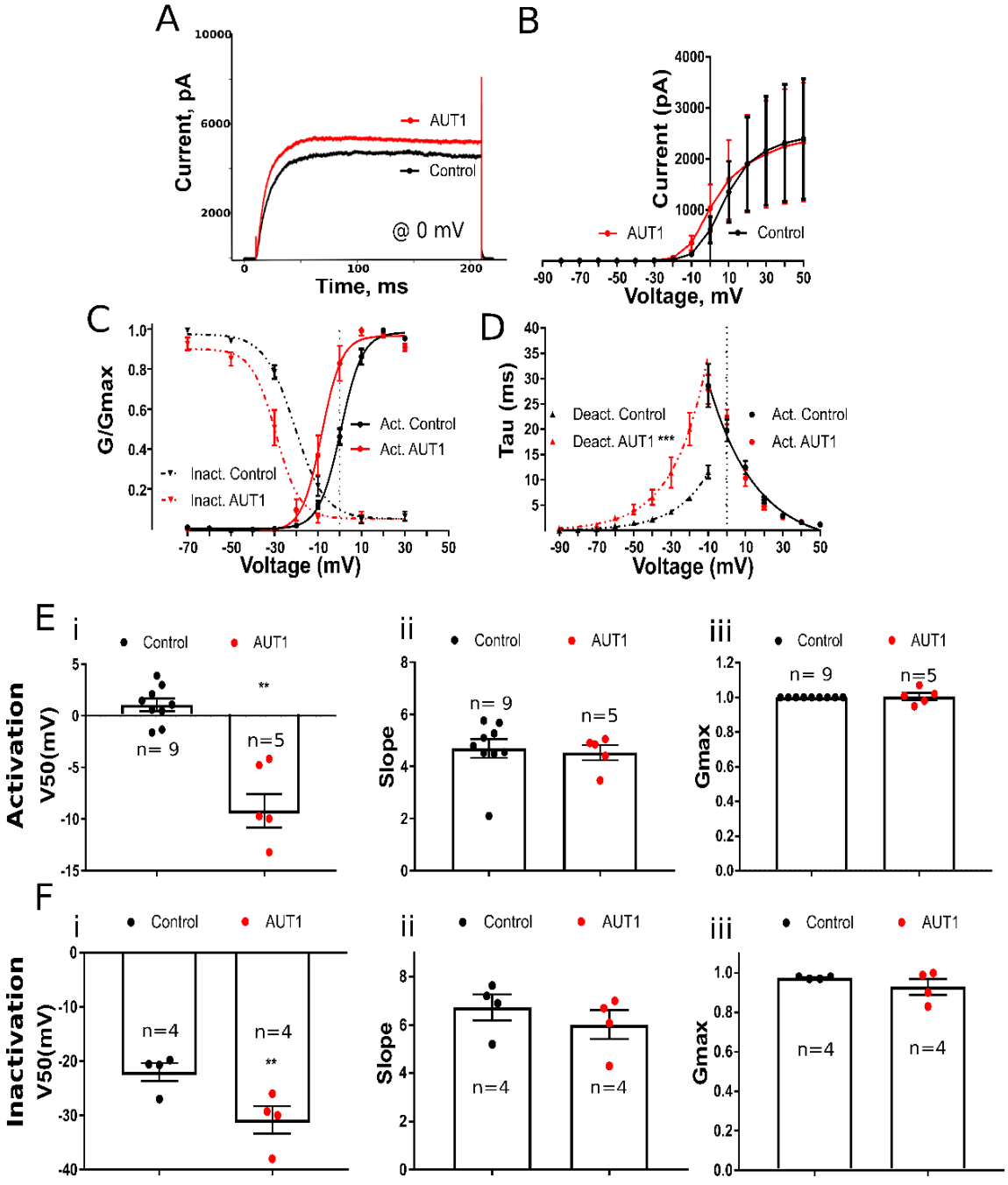

A
Current, pA
AUT1
Control
@ 0 mV
Time, ms
B
Current (pA)
AUT1
Control
Voltage, mV
C
G/Gmax
Act. Control
Act. AUT1
Inact. Control
Inact. AUT1
Voltage (mV)
D
Tau (ms)
Deact. Control
Deact. AUT1 ***
Act. Control
Act. AUT1
Voltage (mV)
E
Activation
i
V50(mV)
Control
AUT1
n= 9
n=5
**
ii
Slope
n= 9
n=5
iii
Gmax
n= 9
n=5
F
Inactivation
i
V50(mV)
n=4
n=4
**
ii
Slope
n=4
n=4
iii
Gmax
n=4
n=4

Figure 3.5. AUT1 slowed deactivation and negatively shifted V50 of Kv3.1b currents.

A) *Representative current trace of Kv3.1b in control (black) and AUT1 (red) at 0 mV.* ***B)*** *Voltage-dependent current in control (black) and AUT1 (red).* ***C)*** *Overlay of normalised Kv3.1b conductances during inactivation and activation protocols in control (black) and AUT1 (red).* ***D)*** *Activation and deactivation tau for Kv3.1b in control and AUT1.* ***E)*** *Comparison of activation Boltzmann parameters, V50, slope and Gmax in control (black) and AUT1 (red) (i, ii, iii).* ***F)*** *Comparison of inactivation Boltzmann parameters, V50, slope and Gmax in control (black) and AUT1 (red) (i, ii, iii). Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.4 AUT1 potentiated peak Kv3.1b current during action potential waveforms but did not alter peak timing

An action potential train at 42 Hz recorded from a spinal neurone was used to create an Axon Text File to be used as the command voltage. This train was recorded at a holding potential of -70 mV in the spinal neurone but mistakenly applied to HEK cells expressing Kv3.1b held at -55 mV, essentially meaning that the peak of the action potentials was 15 mV larger than that of the neurone. This protocol was only used on one Kv3.1b HEK cell, however, the purpose of including this data is to show when the Kv3 current peaks during action potentials and what the effect of AUT1 is during a physiological relevant change in membrane voltage. Using this command train of action potentials, peak Kv3.1b conductance was evidently achieved during the repolarisation phase of an action potential **(Fig. 3.12 A).** Because this experiment was performed on a

single cell results are not discussed in terms of statistical significance. This peak was potentiated with treatment of AUT1 **(Fig. 3.12 B)** and normalisation of the currents confirmed a slower decay during the afterhyperpolarisation and refractory period of an action potential **(Fig. 3.12 C).** Latencies between onset and peak of action potential were unaffected by treatment with AUT1 **(Fig. 3.12 D ii, iii)** however amplitudes and decay tau were visibly increased although multiple repeats would be required to test the significance of this effect **(Fig. 3.12 D iii, iv).** Decay tau in both conditions visibly slowed during the train although in this experiment accommodation during the trains was not controlled for **(Fig. 3.12 D iv).**

In summary, AUT1, negatively shifted the V50 of activation and inactivation of Kv3.1b channels, as well slowed the deactivation kinetics.

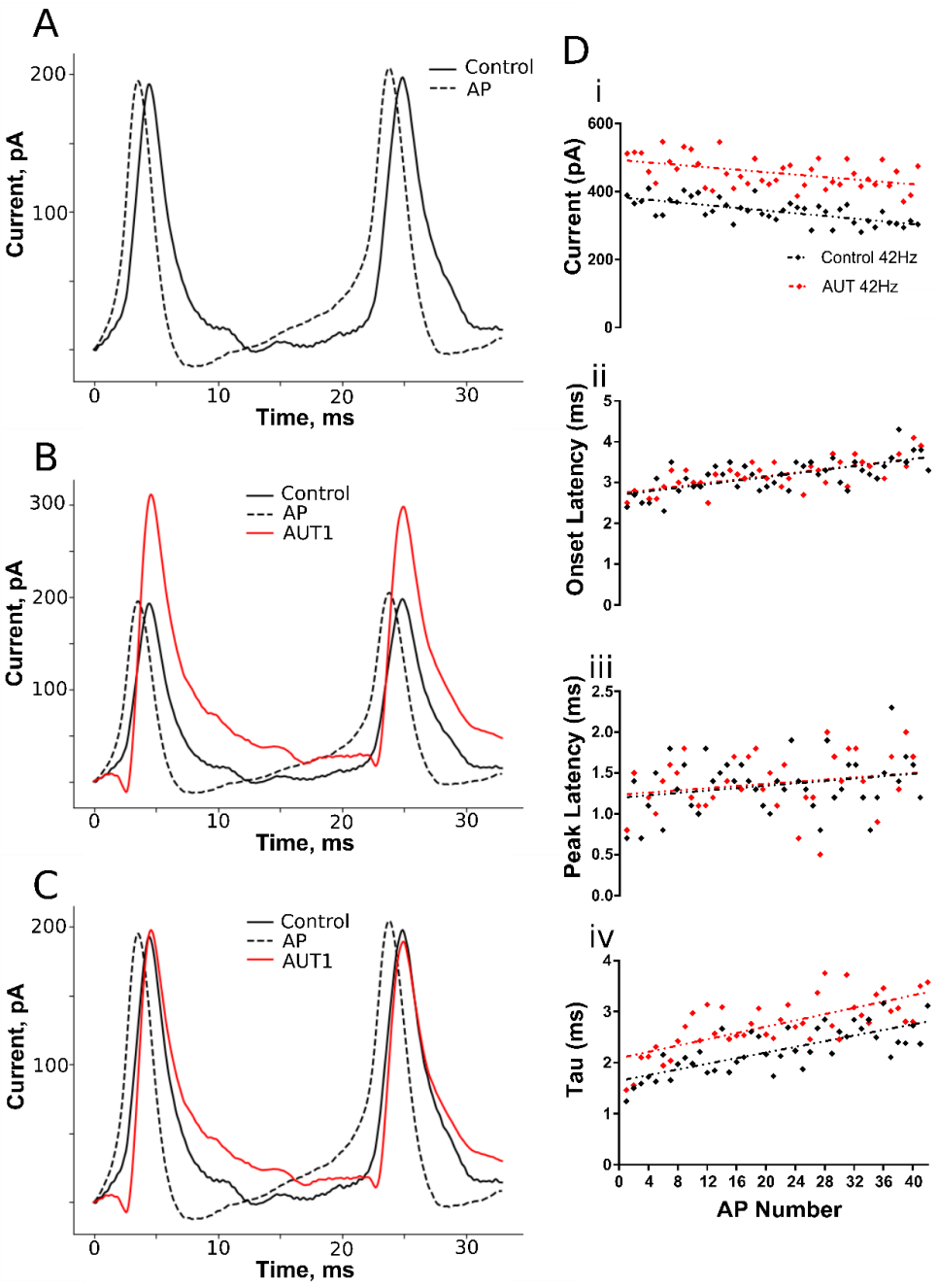


Figure 3.6. AUT1 potentiated peak Kv3.1b current and slowed current decay.

Data from one cell. ***A)*** *Kv3.1b currents (solid line) evoked by a train of action potential waveforms at 42 Hz (dashed line) from a spinal neurone (see Materials and Methods).* ***B)*** *Kv3.1b currents evoked by train stimulation in control (black) and AUT1 (red).* ***C)*** *Kv3.1b currents in control and AUT1 normalised to compare current spike morphology.* ***D)*** *Measures of current amplitude, latency from action potential onset and peak and decay tau of Kv3.1b currents in control and AUT1 (i, ii, iii, iv).*

3.3.5 HEK Kv3.4a transfection produced fast A-type currents

In HEK cells transfected with Kv3.4a cDNA, depolarising voltage steps elicited an A-type current with rapid N-type inactivation **(Fig. 3.13 Ai).** Kv3.4a currents activated around -30 mV reaching maximal conductance at 20 mV **(Fig. 3.13 B).** Inactivation was steep with the channel almost being fully inactivated at -30 mV **(Fig. 3.13B).** Inactivation tau was 8.8 ± 0.09 ms and activation tau was 0.84 ± 0.02 ms at 50mV **(Fig. 3.13C).**

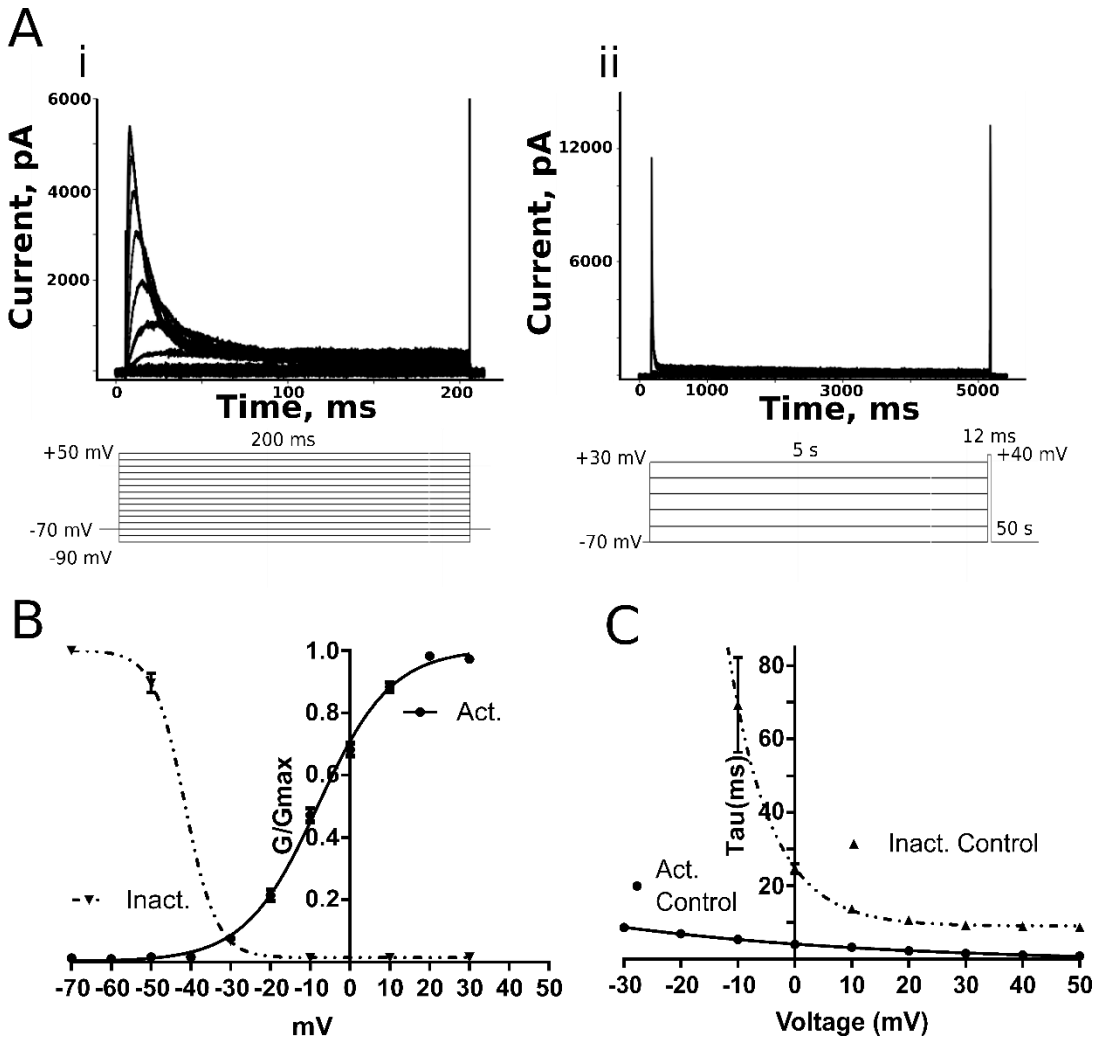


Figure 3.7. Transfection with Kv3.4a cDNA produced rapid A-type current

A) *Representative current traces of Kv3.4a during activation (i) and inactivation (ii) protocols.* ***B)*** *Overlay of normalised conductances and Boltzmann fits during activation (solid line) and inactivation (dashed line) protocols.* ***C)*** *Plot of activation (solid line) and inactivation (dashed line) tau as a function of voltage. Error bars are SEM.*

3.3.6 AUT1 reduced maximal conductance of Kv3.4a channels

In an example current trace during 30 mV voltage steps, Kv3.4a peak current appeared reduced in AUT1 compared to control **(Fig. 3.14 A).** Activation and inactivation was significantly faster in the presence of AUT1 **(Fig. 3.14 D).** Application of AUT1, significantly shifted the V50 of activation by 9 mV (-9.4 ± 1.5 mV to -18.3 ± 1.9 mV, n=9) to more negative potentials and significantly reduced the maximal conductance by 24% (0.97 ± 0.01 to 0.74 ± 0.03, n=9) **(Fig. 3.14 C, F i).** Application of AUT1 also significantly shifted the V50 of inactivation by 7 mV to more negative potentials (-40.6 ± 1.8 mV to -47.6 ± 0.5 mV, n=3) and significantly reduced the maximal conductance by 20% (1 ± 0 to 0.8 ± 0.02, n=3) **(Fig. 3.14 C, Gii).**

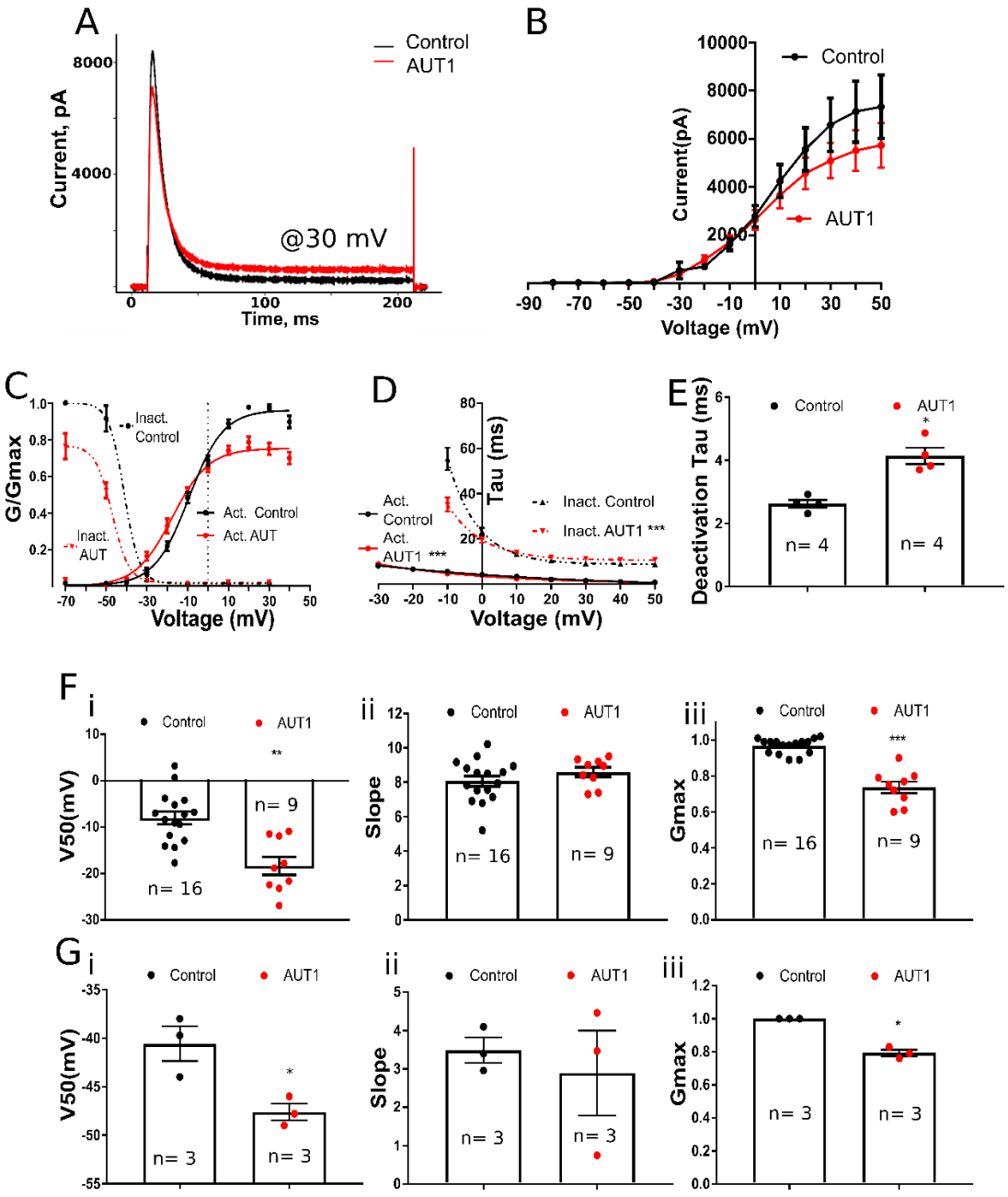

A
Control
AUT1
Current, pA
8000
4000
@30 mV
0
100
200
Time, ms
B
Control
AUT1
Current(pA)
10000
8000
6000
4000
2000
-90 -70 -50 -30 -10 10 30 50
Voltage (mV)
C
G/Gmax
Inact. Control
Inact. AUT
Act. Control
Act. AUT
Voltage (mV)
D
Tau (ms)
Act. Control
Act. AUT1 ***
Inact. Control
Inact. AUT1 ***
Voltage (mV)
E
Deactivation Tau (ms)
Control
AUT1
n= 4
n= 4
F
i
V50(mV)
n= 16
n= 9
ii
Slope
n= 16
n= 9
iii
Gmax
n= 16
n= 9
G
i
V50(mV)
n= 3
n= 3
ii
Slope
n= 3
n= 3
iii
Gmax
n= 3
n= 3

Figure 3.8. AUT1 reduced Kv3.4a maximum conductance and slowed deactivation.

A) *Representative current trace of Kv3.4a in control and AUT1 at 30 mV.* ***B)*** *Current responses to voltage steps in control and AUT1.* ***C)*** *Overlay of normalised conductances and Boltzmann fits during activation and inactivation protocols in control and AUT1.* ***D)*** *Plot of activation and inactivation tau in control and AUT1.* ***E)*** *Deactivation tau obtained from tail currents after 12 ms test pulse in inactivation protocol in control and AUT1.* ***F)*** *Parameters of activation Boltzmann fits in control and AUT1.* ***G)*** *Slope, Gmax and V50 parameters of inactivation Boltzmann fits in control and AUT1. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.7 AUT1 reduced peak conductance during AP waveform stimulation

The same 42 Hz train protocol as in Fig. 3.12 was used on a single Kv3.4a HEK cell and is presented to demonstrate the inactivation of the channel during firing. In this protocol Kv3.4a current peaked during AP repolarisation and appeared to cumulatively inactivate during stimulation **(Fig. 3.15A).** In a separate protocol using a *single* AP waveform applied at the correct holding potential at a rate of 5 Hz, current evoked by each action potential waveform was significantly decreased in the presence of AUT1 compared to control and in both conditions current amplitude decreased as a function of action potential number **(Fig. 3.15 B, Di).** No significant differences were observed in peak latency or onset latency. Despite visual suggestion of a slower decay, when fit with a single exponential function, no significant difference was observed in the decay tau **(Fig. 3.15 D).**

In summary, AUT1 reduced the maximal conductance by increasing the rate of inactivation of Kv3.4a channels, as well as slowing the deactivation kinetics.

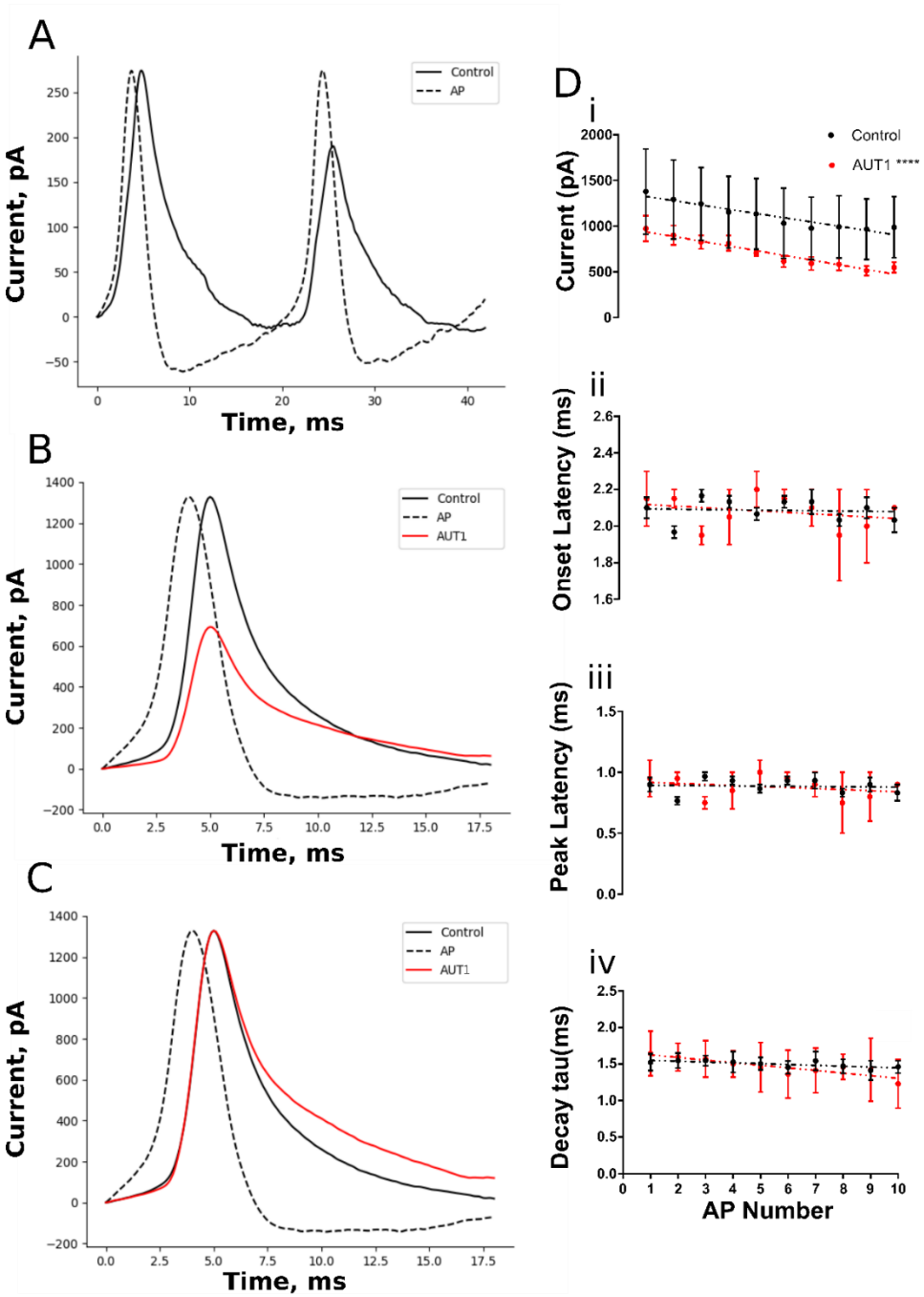

A
Control
AP
Current, pA
Time, ms
B
Control
AP
AUT1
Current, pA
Time, ms
C
Control
AP
AUT1
Current, pA
Time, ms
D
i
Control
AUT1 ****
Current (pA)
ii
Onset Latency (ms)
iii
Peak Latency (ms)
iv
Decay tau(ms)
AP Number

Figure 3.9. AUT1 reduced the peak Kv3.4a current evoked by an action potential waveform.

A,B, C; *Overlay of current evoked by a train of action potential waveforms in control (A) and single waveforms in AUT1 not normalised (B) and AUT1 normalised (C).* ***D)*** *Measures of current amplitude, latency from action potential onset and peak and decay tau of Kv3.4a currents in control and AUT1 (i, ii, iii, iv). Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.8 Co-transfection produced putative heteromeric currents

In an attempt to produce heteromeric Kv3 channels, Kv3.1b and Kv3.4a were co-transfected in the same cells. HEK cells co-transfected with both Kv3.1b and Kv3.4a cDNA exhibited an inactivating current with a steady state component **(Fig. 3.16 Ai).** Kv3.1b/4a currents activated around -20 mV and displayed statistically distinct kinetics of activation and inactivation (determined by fit parameters) compared to Kv3.4a and Kv3.1b alone **(Fig. 3.16 C).** Co-transfection produced currents with a significantly slower inactivation than Kv3.4a alone and fits of activation tau between each channel type were significantly different **(Fig. 3.16 B).** To test the hypothesis that co-transfection produced a distinct channel, current amplitudes were measured at the plateau of the Kv3.1b/4a transient where a Kv3.4a homomeric component is likely to be inactivated. Comparison of activation curves to Kv3.1b homomeric (plateau currents) indicated a different time course of activation suggesting a distinct heteromeric current **(Fig. 3.16 D).**

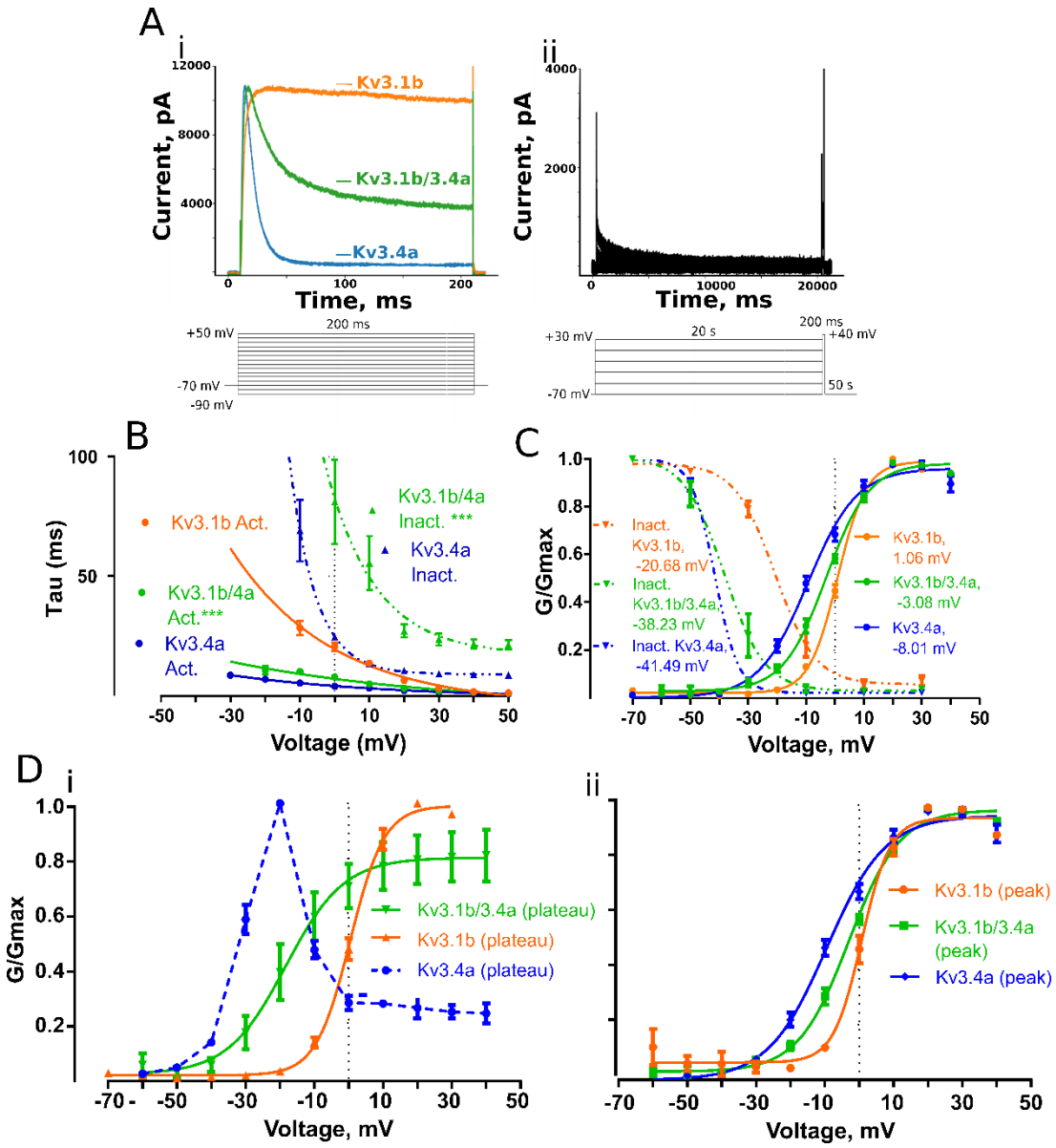


Figure 3.10. Co-transfection of Kv3.1b and Kv3.4a cDNA produced distinct channel kinetics.

A) *Representative current trace of co-transfection of Kv3.1b and Kv3.4a (Kv3.1b/4a, green) in activation and inactivation protocols and in comparison with Kv3.1b(orange) and Kv3.4a (blue) alone.* ***B)*** *Overlay of activation and inactivation curves and Boltzmann fits for Kv3.1b/4a (green), Kv3.4a (blue) and Kv3.1b (orange).* ***C)*** *Plot of activation, deactivation and inactivation taus for Kv3.1b (orange), Kv3.4a (blue) and co-transfection (Kv3.1b/4a, green).* ***D)*** *Comparison of homomeric and heteromeric activation curves measured at the peak (i) and the plateau (ii) current. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.9 AUT1 negatively shifted activation of co-transfection currents

Treatment with AUT1 potentiated Kv3.1b/4a current at 0 mV **(Fig. 3.17 A, C).** Furthermore, the V50 of activation was significantly shifted by 14 mV to more negative potentials (-1.87 ± 0.7 to -16 ± 0.3 mV) and Gmax was significantly reduced by 12% (0.96 ± 0.01 to 0.85 ± 0.2) **(Fig. 3.17 C, Ei, ii).** A single function could describe inactivation and also activation in both control and AUT1 indicating no effect of the compound on this parameters **(Fig. 3.17 D).**

In conclusion, co-transfection produced currents of a channel with distinct kinetics, and AUT1 negatively shifted the V50 of activation and the maximal conductance.

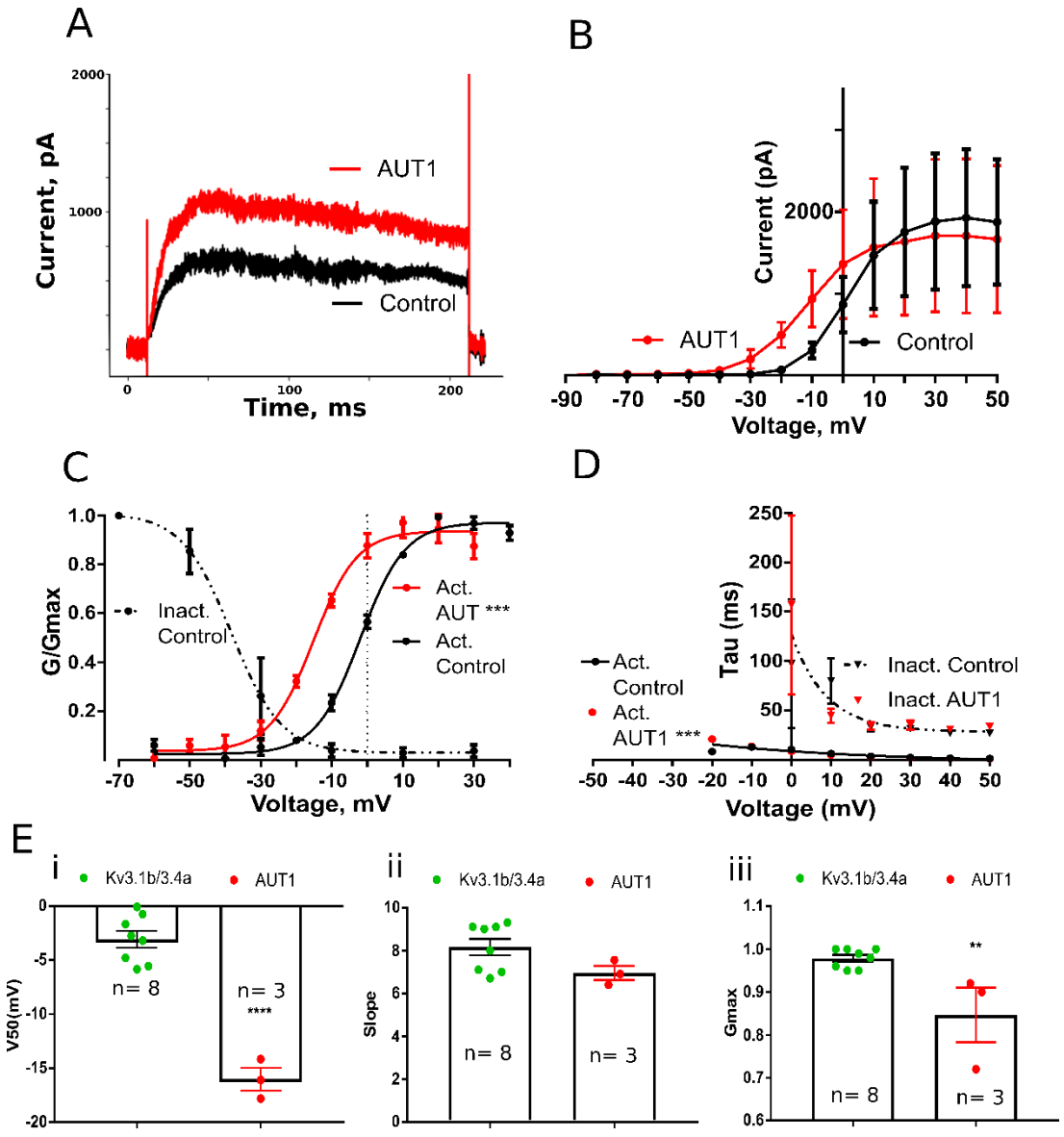


Figure 3.11. AUT1 negatively shifted V50 and reduced Gmax of heteromeric Kv3.1b/4a currents.

A) *Representative current trace of Kv3.1b/4a at 0 mV in control and AUT1.* ***B)*** *Kv3.1b/4a currents evoked by voltage steps in control and AUT1.* ***C)*** *Overlay of activation and inactivation curves and Boltzmann fits for Kv3.1b/4a in control and AUT1.* ***D)*** *Plot of activation, deactivation and inactivation taus for Kv3.1b/4a in control and AUT1.* ***E)*** *Parameters of Boltzmann fits to activation data in control and AUT1. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.10 A two state Markov model suggests AUT1 increased open probability

A simple two state Markov model describing the transition from a closed to open state (see Materials and Methods) was used to fit the activation-deactivation taus **(Fig. 3.18 A)** and conductance **(Fig. 3.18 B)** data for Kv3.1b currents. Simultaneously fitting the control data to Equations 3.7, and 3.8 produced values 0.0217, 0.0324, 0.00254 and 0.00194 for Ko0, Kc0, z0 and zc. For AUT1 these values were 0.032, 0.0128, 0.0024, 0.0016. These parameters were used to solve the rates of opening and closing of the channel in control and AUT1 conditions **(Fig. 3.18 C).** The main effect of AUT1 was evidently a reduction in the closing rate alongside a small increase in the opening rate **(Fig. 3.18C).** The probability of the channel being in an open state is the opening rate as a fraction of the sum of both rates (see Equation 3.8), thus at the intersection the open probability is 0.5. For currents in AUT1 conditions this intersection occurred at more negative potentials and thus indicated a negative shift in the open probability of the channel.

Thus the effect of AUT1 was to negatively shift activation by increasing the open probability of the Kv3.1b channel.

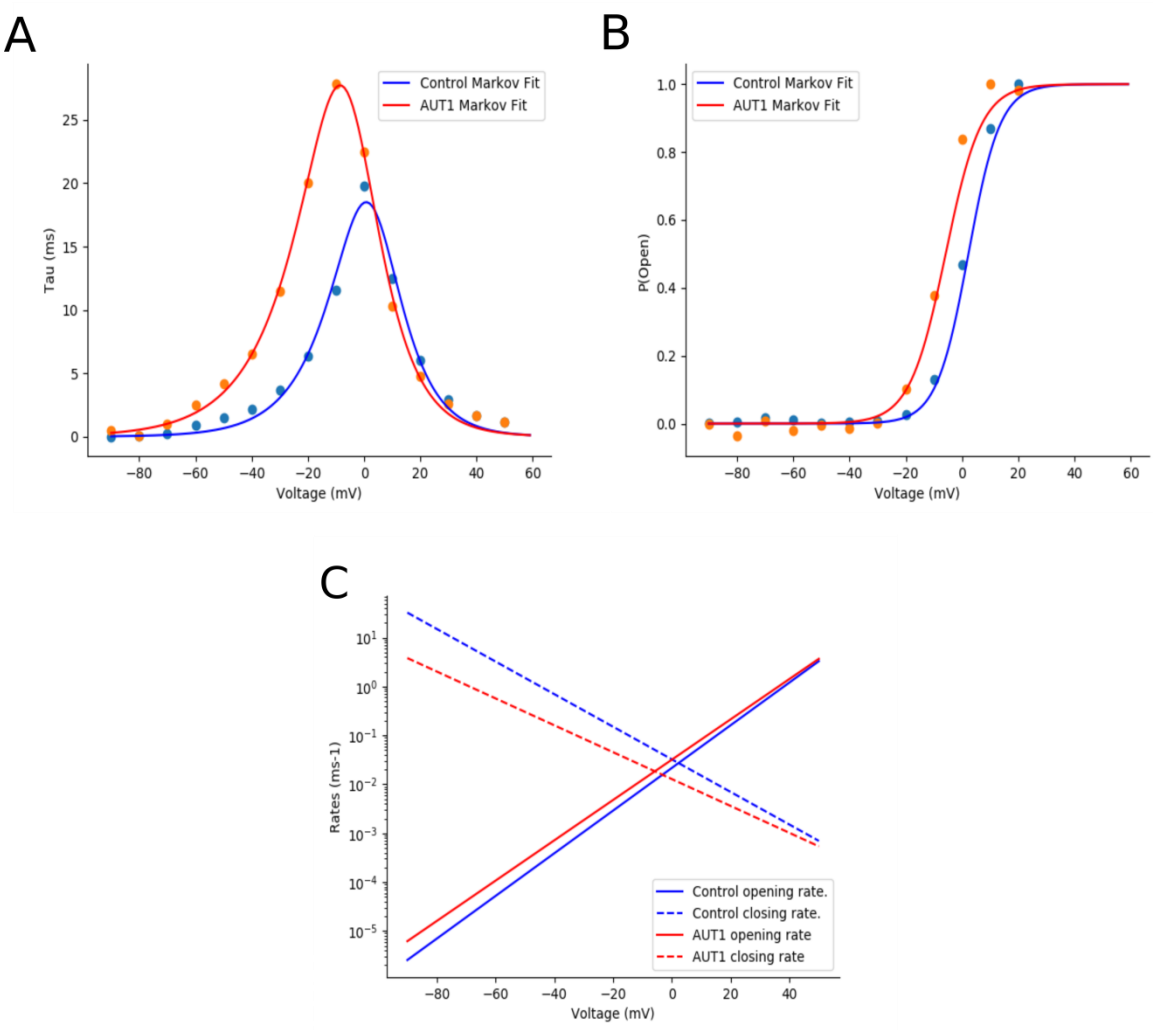


Figure 3.12. A simple two state Markov model describes both Kv3.1b conductance and rate of activation.

A, B*; Activation-deactivation tau-voltage plots (A) and conductance-voltage plots (B) were simultaneously fit to equations determining a two state model (see Materials and Methods).* ***C)*** *Rates of opening and closing derived from parameters solved during the fitting of A and B and plotted to a logarithmic scale.*

3.3.11 Partial recovery of V50 of activation with AUT1 wash off

Wash off experiments to quantify the time taken for the effect of AUT1 on Kv3.1b cells to return back to baseline. One of the main effects of the AUT1 compound on Kv3.1b cells was a negative shift in the V50 of activation (see Fig. 3.11). Plotting this measure over time showed that AUT1 had a rapid onset of action occurring within minutes of bath application **(Fig. 3.19 Aii, Bii).** Upon wash off, V50 was immediately but only partially recovered. This partial recovery was sustained for the wash duration; 10 minutes for one cell **(Fig.19 Aii)** and 30 minutes for another cell **(Fig. 3.19 Bii),** however a full recovery was never attained. This was also represented in conductance-voltage plots showing a partial return to control **(Fig. 3.19 Bi).**

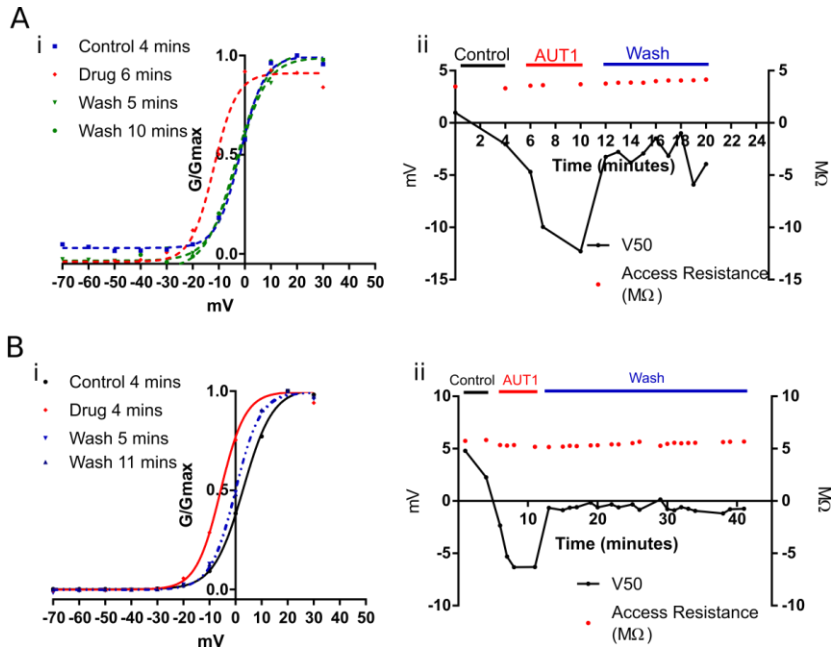


Figure 3.13. Peak AUT1 effect washes off rapidly but partially

The time course of the effect of AUT1 in two separate cells. ***A)*** *A HEK cell transfected with Kv3.1b cDNA with conductance-voltage plots (i) and V50-time plots (ii) during control, AUT1 and wash off conditions. V50 is plotted to the left y axis and access resistance is*

*plotted to the right y axis **B)** Same as A but for a second cell also transfected with Kv3.1b cDNA.*

3.3.12 Can putative heteromeric currents be produced by summation?

Co-transfection of Kv3 cDNA is no guarantee of production of a heteromeric channel. Therefore, 10 different percentages of a Kv3.1b current **(Fig. 3.20 Av)** and Kv3.4a **(Fig. 3.20 Ai)** current were added together to explore the possibility that the currents observed in co-transfection were the sum of two homomeric currents. This produced 100 combinations of currents that were then fitted to single exponentials to quantify inactivation tau and plotted as conductance against voltage to compare Boltzmann fits with fits to the experimental mean. All 100 currents produced conductance curves similar to the experimental mean from co-transfection experiments **(Fig. 3.20 Bi),** however only four combinations were within 20% of the values of the experimental fit parameters **(Fig. 3.20 Bii).** The simulation suggested that conductance curves seen in co-transfection experiments could be reproduced by summing a small proportion of Kv3.4a current with a large proportion of Kv3.1b current. Another key indicator used to identify possible heteromeric currents in co-transfection experiments was the rate of inactivation. Simulated current summations produced an array of inactivation tau **(Fig. 3.20 C)** however a very small proportion of combinations were within 20% of the experimental mean **(Fig. 3.20 C).** The inactivation tau of current summations were evidently skewed to values smaller than the mean **(Fig. 3.20 C).** Comparing the inactivation tau of the four combinations from Fig. 3.19 Bi with the experimental mean showed that the inactivation tau of putative heteromeric currents was reproducible at more depolarised voltage steps **(Fig. 3.20 D iv, v).** However, at relatively more

negative steps this reproducibility evidently dissolved **(Fig. 3.20Di, ii, iii).** This indicated that the currents produced by co-transfection were not the sum of two homomeric channels.

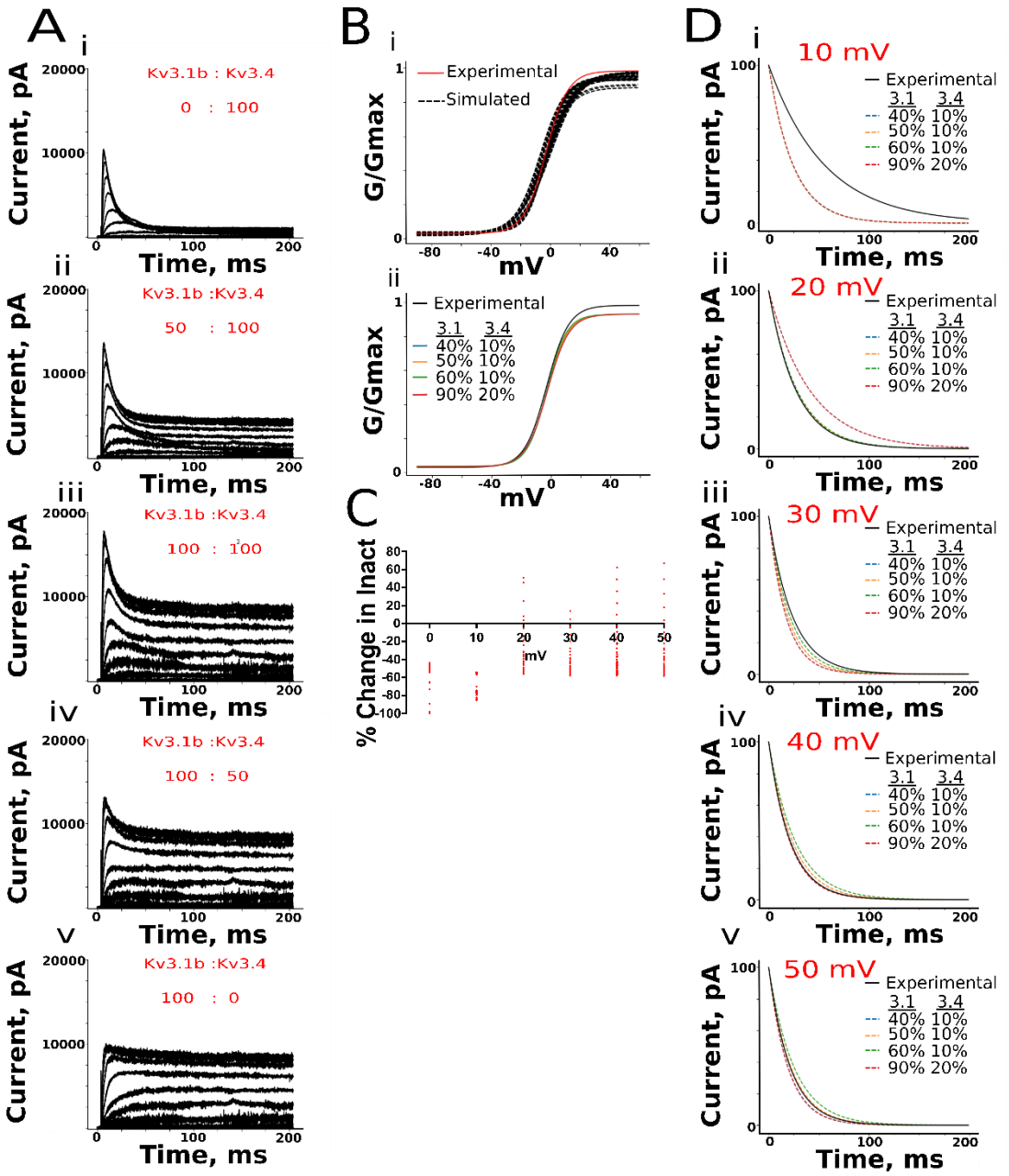


Figure 3.14. Summation of Kv3.1b and Kv3.4a currents reproduces phenotype of Kv3.1b/Kv3.4a co-transfection currents

A) *5 example traces out of 100 of possible currents produced by summation of different percentages of a Kv3.1b current and a Kv3.4a current; 0:100, 50:100, and 100:100.* ***B)*** *Conductance-voltage fits for 100 possible combinations of Kv3.1b and Kv3.4a currents (black dashed lines) compared to the fit of the mean of the experimental data (red solid line) (i). 4 combinations of currents that produced fits with all Boltzmann parameters within 20 % of the experimental fit parameter values (ii).* ***C)*** *100 possible combinations for currents were fit with a mono exponential to describe the rate of*

inactivation(inactivation tau). Results are reported as the percentage change from the experimental fits of inactivation for each voltage step. ***D)*** *Fits of inactivation tau for the four combinations in Bii compared against the experimentally recorded tau at 10, 20, 30, 40 and 50 mV.*

3.3.13 The effect of AUT1 on TEA sensitive lamina VII spinal neurones in the lumbosacral spinal cord.

Spinal neurones in lumbosacral spinal slices were identified in lamina VII in the vicinity of the anatomical location of autonomic motoneurones involved in bladder circuitry and were pre-incubated (>1h) in either control aCSF (0.1% DMSO) or 10 μM AUT1-aCSF (Control=8, AUT=9). Neurones recorded from this region displayed a simple morphology with an axon and only a few main dendritic branches that extended medially, dorsally and ventrally **(Fig. 3.21 A).** Cells were selected for analysis if application of 0.5mM TEA broadened the action potential, increased the peak amplitude and decreased the amplitude of the afterhyperpolarisation as demonstrated in **Fig. 3.21 B.** Estimated input resistance (Rm) was unchanged between cells pre-incubated in control and AUT1 **(Fig. 3.21 C).**

AP and AHP features were measured at rheobase-the minimum current required to evoked an action potential, and firing features were measured at the current required for peak frequency, threshold, duration and peak amplitude of an action potential were unchanged by application of AUT1. Several neuronal features compared displayed trends of change that were statistically insignificant, perhaps due to the low power from the small number of cells recorded in each condition. This included non-significant increases in latency to the first action potential **(Fig. 3.22 A)**, AHP amplitude and duration **(Fig. 3.22**

B), and decreases in firing frequency and interspike interval coefficient of variation **(Fig. 3.22 C).** Only the effect on instantaneous firing frequency reached statistical significance ($p<0.05$) **(Fig. 3.22 C)** but it would be interesting to see if these trends become significant with greater statistical power. In conclusion, AUT1 reduced the firing frequency but a greater statistical power would be required to identify any other significant changes.

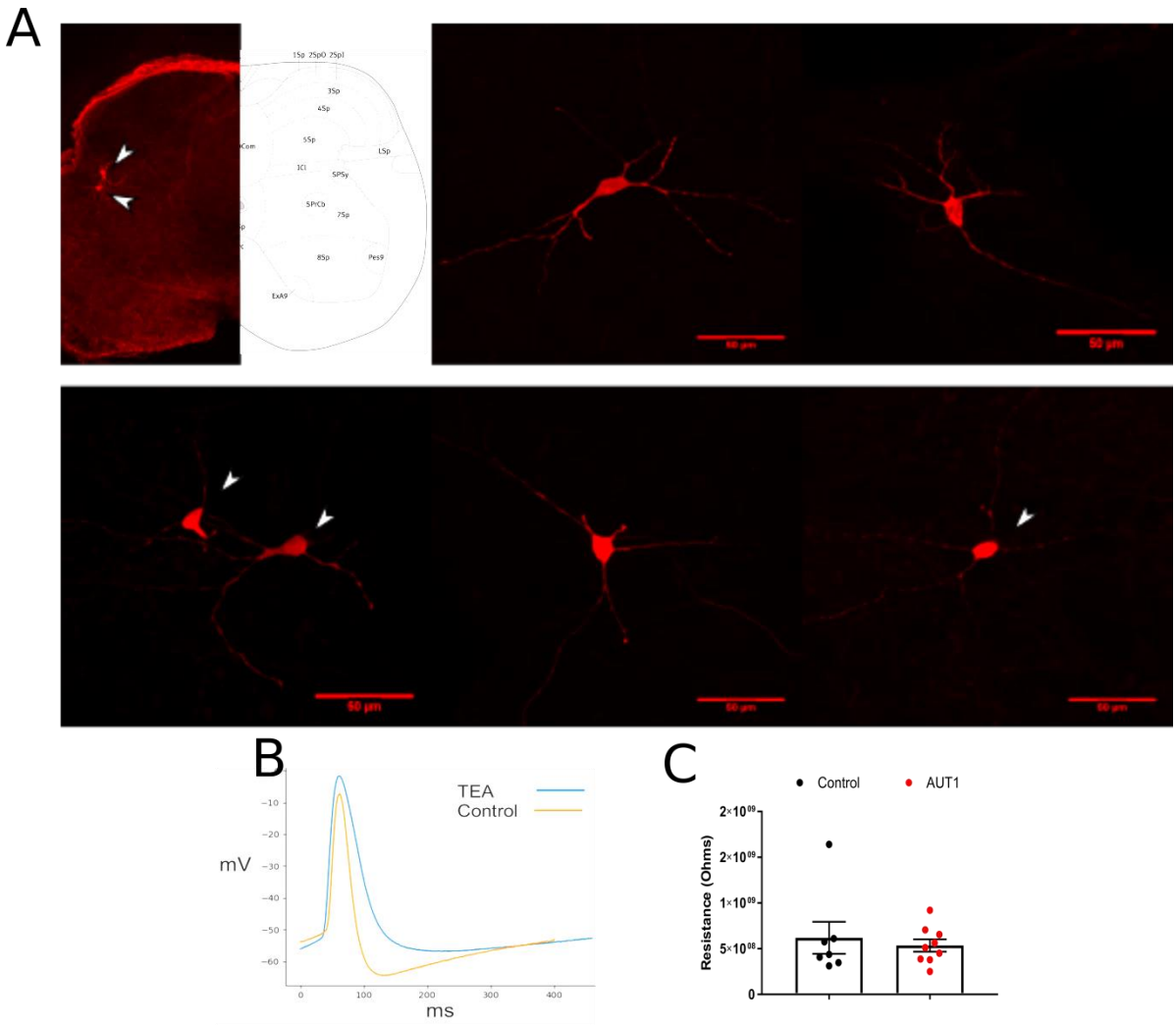


Figure 3.15. Spinal neurones in lamina VII were sensitive to TEA.

A) *Biotin-filled TEA sensitive neurones (white arrows) located in lamina VII of the lumbo-sacral spinal cord.* ***B)*** *An example trace showing that application of TEA broadened AP width, reduced AHP and increased amplitude.* ***C)*** *Input resistance of TEA sensitive cells for both control and AUT1 conditions.*

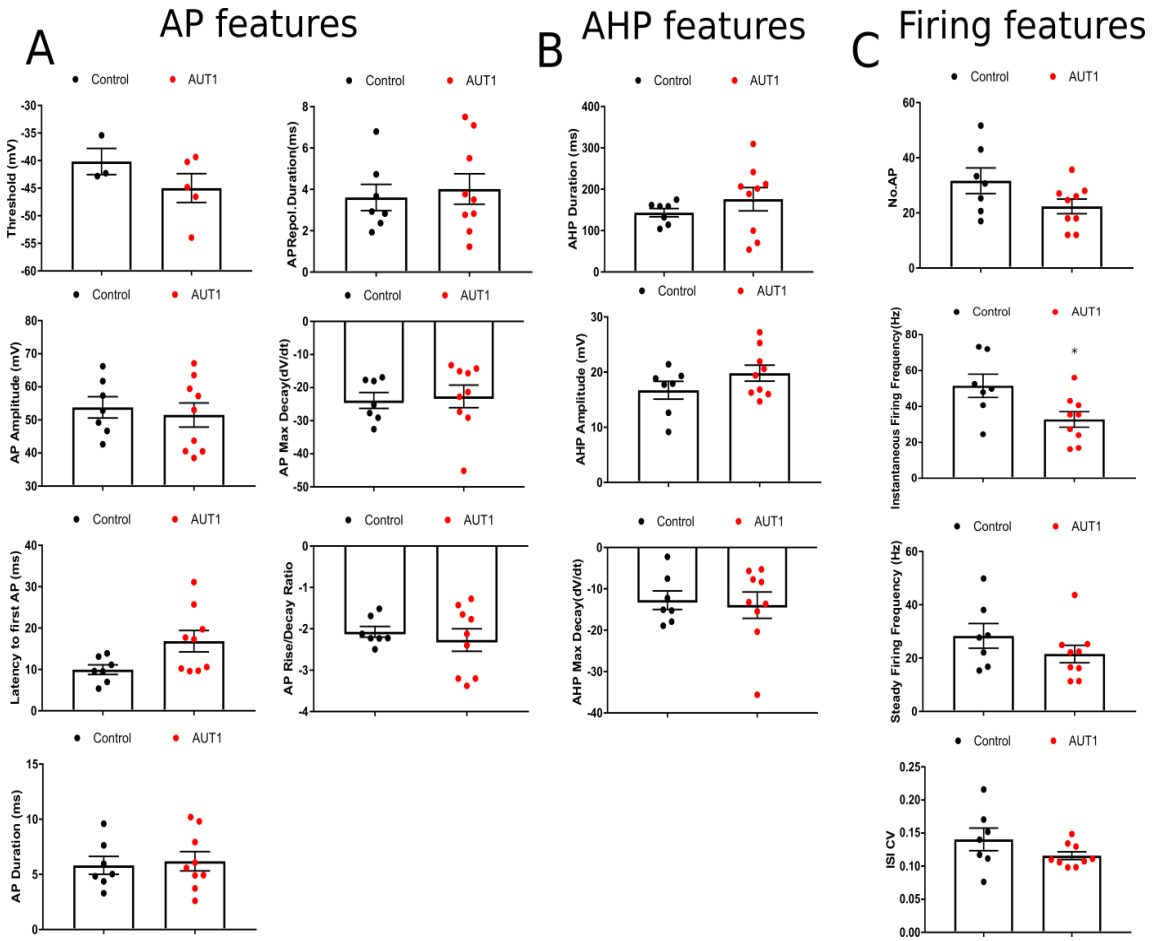


Figure 3.16. AUT1 reduced instantaneous firing frequency of TEA sensitive neurones.

A) *Features of the action potential waveform such as duration, amplitude, rates of rise and decay were measured.* ***B)*** *Features of the afterhyperpolarisation such as amplitude and duration were measured.* ***C)*** *Frequency of firing was measured, instantaneous firing frequency was estimated from the first interspike interval (ISI), whereas the steady firing frequency was estimated from the last five ISI. A coefficient of variation was calculated for the ISIs (ISI CV). Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

3.3.14 Action potential analysis during maximal firing

The slower deactivation of Kv3 channels in AUT1 could be hypothesised to sustain AHPs for longer during a period of more intense activity such as a train of action potentials. In this scenario, it would be important to analyse AP and AHP features during such a train. During current stimulation that evoked the maximal firing frequency, the second, fifth and tenth AP and AHP were analysed **(Fig. 3.23).** For cells incubated in AUT1 AP threshold and peak amplitude were unchanged **(Fig. 3.23A, B).** Interestingly, the AP duration and repolarisation duration were unchanged **(Fig. 3.23C, D).** The most evident difference was a statistically significant increase in the duration (Control=14.4 ms, AUT1=30 ms, $p<0.05$, for the second AP of the train) **(Fig. 3.23E)** and amplitude (Control=9 mV, AUT1=15.9 mV, $p<0.05$, for the second AP of the train) **(Fig. 3.23F)** of the AHP most likely due to a slower deactivation of Kv3 channels and that most likely underlined a reduction in instantaneous firing frequency **(Fig. 3.23F).**

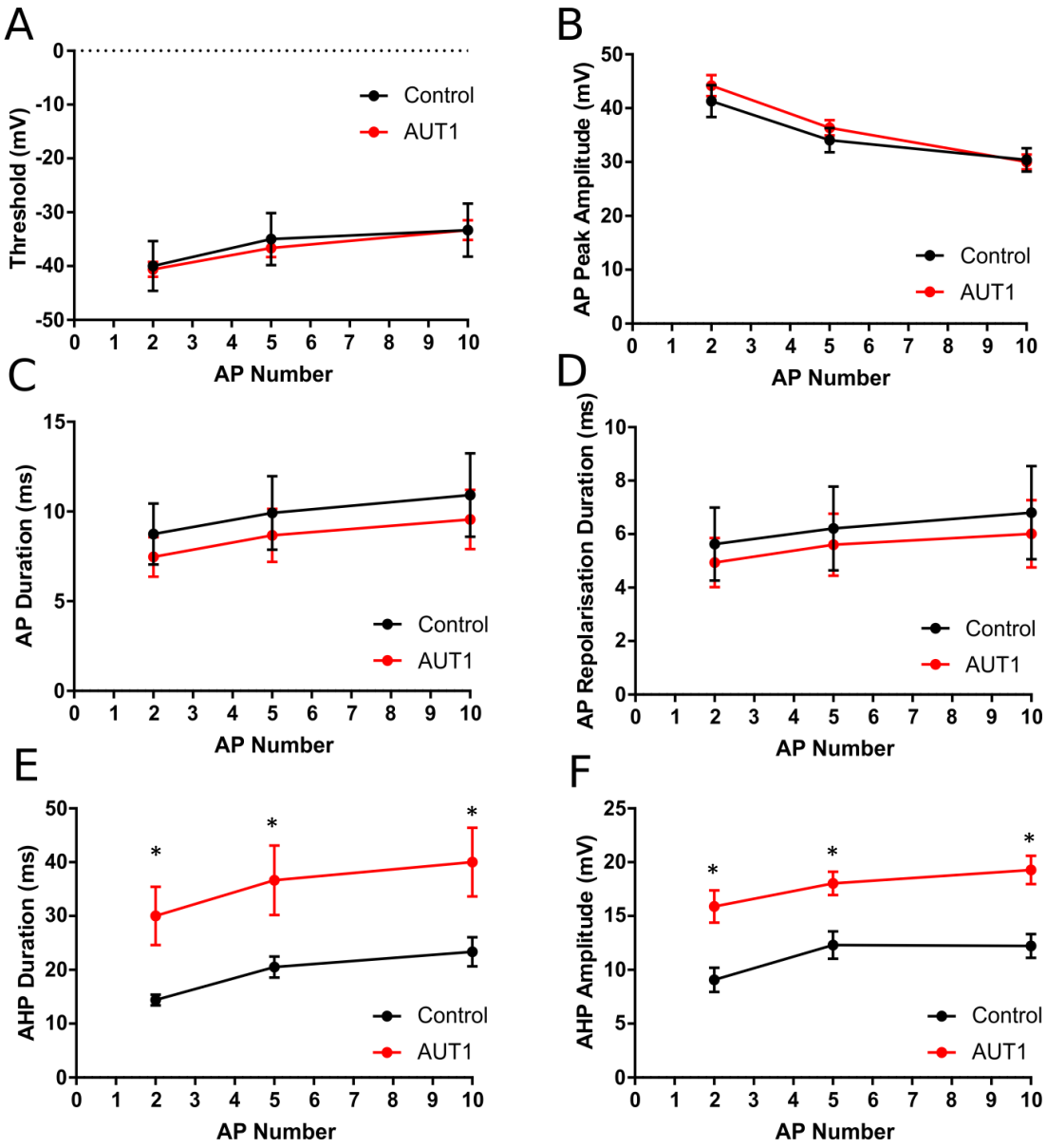


Figure 3.17. AUT1 significantly increased the AHP amplitude and duration of multiple APs during a firing train.

***A-F**; Comparison of AP threshold **(A)**, amplitude **(B)**, duration **(C)**, repolarisation **(D)** and AHP duration **(E)** and amplitude **(F)** for several APs during a train of firing (>20Hz) between independent TEA-sensitive neurones pre-incubated in either control (n=8) or AUT1 (n=9). Errors represent SEM.*

3.4 Discussion

3.4.1 Using the GHK equation instead of Ohms law to calculate conductance

Because K+ currents with an asymmetric distribution are rectified by both the chemical and electrical gradient, they cannot be accurately described by a linear ohmic relationship and thus can be described by GHK normalisation (Clay, 2000). The implications of the GHK normalisation were a steepening and negative shift of the voltage dependence. This approach indicated that Kv3 channels are fully activated earlier than modelled using Ohms law. This is crucial for developing accurate models of neurones that contain these channels and should be considered in the development of such models.

3.4.2 Does co-transfection result in a heteromeric channel?

The main limitation of this study is the uncertainty in stoichiometry of a channel produced by co-transfection of Kv3.1b and Kv3.4a subunit cDNA. However, several observations suggest that the currents recorded predominately originate from a distinct heteromeric channel. Firstly, Boltzmann fits of normalised conductances as a function of voltage in activation and inactivation experiments differed from homomeric channels. Likewise kinetics of activation and inactivation were markedly different; Kv3.1b/4a heteromers inactivated and activated slower than Kv3.4a homomers. Despite this and to investigate whether the recorded heteromeric currents were simply the sum of Kv3.1b and Kv3.4a heteromeric currents, different ratios of Kv3.1b and Kv3.4a homomeric currents were summed, normalised conductances calculated and rates of inactivation fitted. In these simulations, summation, typically of a large proportion of Kv3.1b to a small proportion of Kv3.4a, was able to reproduce

many of the features of the putative heteromeric currents, indicating that there is a real possibility that the heteromeric currents are the summation of two homomeric currents. However, whilst many features were recapitulated, some such as the rate of inactivation at less depolarised steps, were not. Furthermore, measuring the plateau of Kv3.1b/3.4a heteromeric currents, a point at which any homomeric Kv3.4a currents would be almost fully inactivated, was unable to follow the fit of Kv3.1b plateau currents from homomeric cell lines. In addition, work by Baranauskas *et al.*, (2003), performing similar co-transfection experiments with Kv3.1b and Kv3.4a cDNA demonstrated co-precipitation of the two subunits and a channel with distinctive properties at the single channel level. But because there is evidence for and against, the inconclusiveness maintains the limiting nature of co-transfection to produce and accurately interpret heteromeric currents.

3.4.3 What are the implications of heteromeric assembly?

Here, putative heteromeric assembly in HEK cells, produced a current that activated ~10mV earlier than homomeric Kv3.1b channels. The significance of this is that during an action potential a Kv3.1b/3.4a heteromer is more likely to activate and provide efficient repolarisation of the neuronal membrane than a Kv3.1b homomer. Indeed, in fast-spiking cortical neurones containing both Kv3.1b and Kv3.4a mRNA, BDS-I (a Kv3.4 blocker) broadened the action potential and reduced the firing frequency indicating that Kv3.4 subunits were important in the repolarisation efficiency of these neurones (Baranauskas *et al.*, 2003). Interestingly, these neurones exhibited a delayed rectifier current

reminiscent of a Kv3.1b/4a heteromer with a small transient decay and an activation voltage more negative than a homomeric Kv3.1b channel.

In addition to shifting the activation voltage to more negative potentials, heteromerisation with Kv3.4a subunits clearly endows a channel with N-type ball and chain inactivation. The rate of inactivation is proportional to the number of inactivating particles (or ball and chains) with more subunits endowing a greater inactivation within a heteromer. As a homomer, during repetitive stimulation by action potentials or during a train of action potentials, Kv3.4a currents cumulatively inactivated. So, it would be expected that the functional effect of cumulative inactivation of Kv3.4a homomeric or heteromeric channels would be action potential spike broadening and a slowing of firing frequency as sustained firing progressed, akin to block with BDS-I.

In extension of this, a further implication of Kv3.4a integration into a Kv3 heteromer, is in a relatively recently reported form of presynaptic plasticity (Rowan and Christie, 2017). In cerebellar interneurones, subthreshold depolarisations at the soma that spread into the axon and synaptic boutons broadened the synaptic action potentials and increased neurotransmitter release. The authors proposed that these subthreshold depolarisations inactivated Kv3.4 subunits reducing the available Kv3 current normally critical in maintaining brief action potentials at boutons. While this is compelling and probably the case during high activity (see Kv3.4a cumulative inactivation above), subthreshold depolarisation (in these neurones threshold was ~ -45 mV) is in contrast to the idea in recombinant cell lines that Kv3.4 subunits are activated ~-30 mV during action potentials, and that inactivation is activation

dependent (Fineberg et al, 2012) and also occurs ~30 mV. It's unknown whether this plasticity is mediated by Kv3.4 homomers or heteromers or a mix of both. However, it is very possible that in native neurones, inactivation of Kv3.4 homomers and heteromers containing Kv3.4 is negatively shifted by interactions with other proteins.

3.4.4 How do these results compare with other studies?

Several other studies have investigated the effect of AUT1 on Kv3 channels, predominately focussing on Kv3.1 channels, (Rosato-Siri *et al.*, 2015; Taskin *et al.*, 2015; Brown *et al.*, 2016; Boddum *et al.*, 2017) but how do the results described here compare? Two of the main findings here were a negatively shifted activation curve and slower deactivation kinetics. For the activation curve, Rosato-Siri *et al.*, 2015 reported that 10 μM AUT1 produced a 4.6mV negative shift in V50 of activation and a 5.9 mV negative shift in V50 of inactivation in human Kv3.1b (hKv3.1b) channels expressed in CHO cells, consistent with an 8.8 mV and 8.6 mV negative shift, respectively, seen here in hKv3.1b in HEK cells. Interestingly, in data presented here the negative shift in both activation and inactivation were almost identical, perhaps indicating that the potentiated current in the inactivating pre-steps shifts the voltage dependence of inactivation. Concluding whether this is a negative shift in inactivation or an experimental artefact due to potentiated pre-steps currents is impossible to dissect here. In general agreement Taskin *et al.*, 2015 reported a 14.63 mV negative shift in V50 of activation in HEK cells and Brown *et al.*, 2016 a 17.2 mV and 3.88 mV negative shift in V50 of activation and inactivation, respectively, in rat Kv3.1b expressed in CHO cells.

In terms of activation and deactivation kinetics, Brown *et al.*, 2016 reported that the rate of activation at -15 mV and deactivation in general was slower in AUT1 compared to control, whereas in data presented here deactivation was significantly slowed but no difference in fits of the rate of activation was observed. Interestingly, Taskin *et al.*, 2015 recorded faster activation kinetics in Kv3.1a in HEK cells with a higher 30 uM dose of AUT1 (RE1 in Taskin et al. 2015) only at 0 mV and 10 mV but again also observed a consistent slowing in deactivation kinetics. The same group when investigating the selectivity of AUT1 (RE01) on Kv3 subunits in oocytes, Boddum *et al.*, 2017 found that 30 uM AUT1 increased the rate of activation by 50% on Kv3.1b channels in addition to a 13.3 negative shift in V50 of activation. Curiously, no effect of the compound on Kv3.4a channels was observed even at higher doses, however a similar compound in the study, EX15, reduced the Kv3.4a current by 50%. No data on activation or deactivation of Kv3.4 channels was reported in this study, however, in the present study, measuring the tail currents of Kv3.4a channels after a very short test pulse in inactivation protocols revealed a deactivating current with a slower rate of decay in AUT1 compared to control. This suggests that a slowed deactivation is a common effect between two distinct Kv3 subunits, Kv3.1b and Kv3.4a.

3.4.5 Slowed deactivation- a common mechanism to explain the contrasting response in Kv3.1b and Kv3.4a channels?

Application of AUT1 reduced the maximal conductance of Kv3.4a channels yet also potentiated the conductance of Kv3.1b channels at several potentials. To unify these differing actions, I propose that slowed deactivation kinetics, an AUT1-induced effect common to both channels, would favour an open state and

produce potentiation in Kv3.1b channels but increase the rate of inactivation in Kv3.4a channels, as it is open-dependent (Fineberg, Ritter and Covarrubias, 2012), to in turn reduce conductance. Indeed, this increase in open probability is evident in the potentiation of early currents (~-20 mV) and the activation curve shift of both homomers. Furthermore, the rate of inactivation of Kv3.4a at -10 mV was significantly faster in AUT1 compared to control. Kv3.1b channels undergo rapid relaxation of the voltage sensing domain (VSD) before channel opening, forcing channels into a pre-active-relaxed state (Labro *et al.*, 2015) which under short depolarisations slows down deactivation kinetics producing a resurgent current as the activation-deactivation balance is disrupted. Is it possible that AUT1 induces a similar duration-independent slowing of deactivation through relaxation and slowing of VSD deactivation? In this delayed deactivation hypothesis AUT1 may bind the channel and either favour relaxation of the VSD or deceleration of return of the VSD from relaxation, increasing the open probability, producing a negative shift in the V50 of activation that increases the rate of Kv3.4a inactivation. In essence the effect of AUT1, by slowing the rate of deactivation, could be to increase the amount of time that channels populate a pre-active state. As entering an open state from a pre-active state is favourable, an increased population of pre-active Kv3 channels would lead to an increase in open probability.

3.4.6 AUT1 on Kv3 gating-inferences from other gating modifiers

The gating of Kv channels is determined by the movement of the voltage sensing domain (VSD) of each subunit in response to changes in the membrane potential. The VSD moves outwards during activation and inwards during

deactivation and when all VSDs are outwards, a concerted movement to open the S6 gate occurs. The effect of AUT1 on Kv3 channels is evidently as a gating modifier however the binding site is unknown. By examining other known gating modifiers we can postulate potential sites of action. Gating modifiers are typically naturally occurring toxins that can be small and lipid-soluble acting within the membrane or large polypeptides that act on extracellular sites of action often to trap the movement of the VSD (Catterall *et al.*, 2007). Because the AUT1 compound has been reported as lipophillic (Brown *et al.*, 2016), we can rule out such extracellular sites of action such as that of BDS toxins (Wang, Robertson and Fedida, 2007), and assume an intra-membrane site of action.

The toxin gambierol is lipophilic too but acts to inhibit Kv3.1 channels by binding the lipid exposed surface of the pore domain (S5-S6) and anchoring the channel in the closed state (Kopljar *et al.*, 2009, 2013). This shifts the activation voltage of the channel significantly to more depolarised potentials. The authors proposed that the site of action could be homologous to site 5 in NaV channels where toxins act as allosteric modulators to favour the open state of the sodium channel. Perhaps, the effect of AUT1 is the opposite of gambierol and homologous to that of NaV site 5 toxins, where AUT1 may bind the gambierol site during activation with a slow dissociation that slows closing of the S6 gate and thus slows deactivation. But is there any evidence to suggest that the affinity of AUT1 is voltage-dependent? Interestingly, the slower deactivation of Kv3.1b in AUT1 was most prominent in the range between -60 mV and -10 mV, returning to control values at -80 mV. This indicated that the effect on deactivation was voltage -dependent and that perhaps the affinity of AUT1 was

voltage-dependent with a slower dissociation and thus greater affinity at depolarised but negative potentials. Obviously other hydrophobic sites within the Kv3 channel may be the site of action such as within the VSD but the idea itself that AUT1 binds with highest affinity during activation and dissociates slowly is a plausible mechanism for the gating modifying compound action. A further consideration is the inability to fully wash off the effect of AUT1 on the V50 of activation for Kv3.1b. Whether this is due to a deep site of action within the membrane, a slow dissociation constant or other properties of AUT1 binding is unclear, but it highlights a lack of complete reversibility of the compounds effect within the 30 minute time frame analysed. Another consideration is the effect of intracellular dialysis on the activation curve. Control recordings of the same duration in the absence of AUT1 would be required here to unpick this partial recovery.

3.4.7 The implications on neuronal firing

The present study found in lumbo-sacral TEA sensitive spinal interneurones that pre-incubation of acute spinal slices in AUT1 resulted in a trend of reduced firing frequency, interspike variation and increased duration and amplitude of afterhyperpolarisations (AHP) at rheobase. Given the hypothesis that slowed deactivation underlies the mechanism of AUT1 action, these results would be expected, however only the reduction in instantaneous firing frequency was significant. Interestingly, analysing AP waveforms at higher firing frequencies showed that cells pre incubated in AUT1 had larger and longer AHPs. An important facet of normal Kv3 kinetics is that they deactivate rapidly to allow the membrane to depolarise again to spike initiation. Therefore, an increase in

Kv3 current due to slow deactivation in the afterhyperpolarisation phase would drive the membrane potential down to more negative potentials and slow the onset to the next action potential reducing the ability of the neurone to fire at high frequencies. Elegant experiments using dynamic clamp control of Kv3 conductance in fast-spiking hippocampal interneurones (Lien and Jonas 2003) strongly support this hypothesis. Despite being reported over a decade ago, many of the modifications made to the conductance of Kv3 channels happen to recapitulate the effect of AUT1 on Kv3 channels, namely slowed deactivation kinetics and a negatively shifted V50 of activation. This group showed that slowing deactivation of Kv3 conductances increased the regularity of firing, and decreased the firing frequency by increasing the AHP amplitude, similar to that observed here. Conversely, accelerating deactivation led to a low frequency irregular firing pattern due to insufficient recovery of sodium channels from inactivation. Shifting the V50 of activation by -10 mV from -12.4 mV produced little evident change in firing frequency, however a shift by -20 mV increased the threshold required for fast spiking and a shift of -30 mV produced a strong adaptation and reduction of firing frequency. Accordingly, AUT1 would have to shift the V50 of Kv3 channels by -30 mV to force it to behave like a threshold setting LVA Kv1 channel, as previously suggested (Olsen et al, 2018, Brown et al, 2016). Ultimately, by injecting kinetically-altered Kv3 conductances into fast-spiking hippocampal interneurones, these authors have laid the groundwork for understanding the effect of AUT1 on neuronal behaviour.

Interestingly, other studies into the effect of AUT1 on neuronal firing have reported a mix of results, with differing hypotheses. One hypothesis, already

alluded to, is that the negative shift in activation makes the Kv3 channel behave like a low-voltage-activated (LVA) channel that often sets the threshold and suppresses excitability (Brown *et al.*, 2016). As a result, threshold should be increased and action potential amplitude and width reduced. I would argue here that the shift in activation voltage would not be significantly large enough for it to behave like a threshold setting LVA K+ current that activate at around -60 mV (Brew and Forsythe, 1995; Dodson, Barker and Forsythe, 2002). Furthermore, AP waveform experiments presented here and in Boddum *et al.*, (2017) suggest that AUT1 doesn't shift peak Kv3.1b conductance to earlier in an action potential waveform, but only increases amplitude and slows current decay. However, using a different current clamp protocol to that used in the present study, with 0.3 ms current pulses instead of a long depolarising current step, they found results to support their hypothesis. The amount of current required to stimulate an action potential (termed AP threshold but a bit of a misnomer) was increased from 2 nA to 2.5 nA (n.b. this seems like quite a high amount of stimulation and may reflect the inability to fully charge the capacitance of the pipette with such a brief pulse) and AP amplitude was reduced by ~10 mV with no effect on AP width. Repetitive stimulation with these short bursts resulted in a reduced number of action potentials with AUT1 treatment. Another experimental difference is that these appear to be paired recordings however no measures of run-down or series resistance are reported over the duration of the experiment. On the other hand, unpaired recordings, as in the present study, also have their limitations. Other unpaired studies have looked at AUT1 in a “recovery” scenario, where a proportion of Kv3 channels are blocked by 1mM TEA (Rosato-Siri *et al.*, 2015; Olsen *et al.*, 2018) and whilst

finding no significant effect of the compound alone, some features of control neuronal behaviour such as firing rate were recovered. Finally, in fast-spiking interneurones from rat hippocampal slices, Boddum et al., (2017) found an increase in the number of evoked action potentials during both “weak” and “strong” step depolarisations.

In agreement with data presented here, a putative potentiation of Kv3 currents during the repolarisation phase of a spinal neuronal action potential coupled with a slower deactivation would lead to a deepening and lengthening of the AHP. However, the subtle determinant of this is the minima of the AHP, for the difference between minima that reach -50 mV and that reach -70 mV is that the slowing in deactivation is greater at -50 mV than -70 mV. Therefore the more depolarised a minima the more the frequency of firing would be theoretically attenuated by available potassium current, whereas the more hyperpolarised the minima the more likely the firing frequency is increased as the normal function of Kv3 channels is potentiated. While channel conductance changes membrane voltage, it is also intrinsically determined by said voltage, therefore, an increase in AHP amplitude by AUT1 may self-regulate the Kv3 channel by driving the membrane down to a potential at which the channel closes still quickly and this may explain the subtle effect on neuronal firing. However, in conditions where the cell cannot hyperpolarise to very negative potentials, such as during a large current step, an increase in AHP duration would be observed. Perhaps, there exists a difference in AHP minima between cells and protocols recorded in the cortex, hippocampus, auditory brainstem and spinal cord that may explain the variable effects of AUT1 on AP shape and firing.

Chapter 4

4 The effect of Kv3 modulation on bladder output

4.1 Introduction

4.1.1 The role of Kv3 channels in bladder function

Bladder function is controlled by a spinal-pontine micturition reflex (see General Introduction). Chapters 1 and 2 indicated that Kv3 subunits are localised to synapses in regions of the spinal cord associated with the micturition reflex and that AUT1 could modulate the properties of homomeric and heteromeric ion channels and lumbosacral TEA sensitive spinal interneurones. In these neurones AUT1 altered their excitability, and this alteration in interneuronal excitability that may be antecedent to bladder motoneurones could also affect the output of bladder motoneurones themselves to the bladder. It was therefore important to investigate the central effects of AUT1 on bodily functions controlled by the spinal cord, such as bladder function. By using a selective compound it would be possible to delineate the role of Kv3 channels in bladder function.

In addition, in humans, bladder function becomes dysregulated with age and it has been postulated that some of this dysregulation occurs at the level of central synaptic loss observed in rodents (Merican 2016, unpublished). Chapter 1 of this thesis indicated that the excito-inhibitory balance of synapses onto bladder motoneurones is altered in age and that a general loss of Kv3-positive synapses is observed. Indeed, the AUT1 compound has often been examined and observed to recover function in conditions designed to simulate loss of Kv3

channels by addition of TEA (Rosato-Siri *et al.*, 2015; Olsen *et al.*, 2018). We therefore postulated that the AUT1 compound may be able to recover normal Kv3 function in the spinal cord and perhaps positively affect bladder function in aged mice.

4.1.2 Aims

- **To investigate whether modulation of Kv3 channels using AUT1 affects bladder function in young and aged mice.**

4.2 Materials and Methods

4.2.1 Micturition

Female C57/bl6 mice (3 month, n=8; 18 month, n=8; 20 month, n=4) were acclimatised to metabolic cages for 3 four hour sessions prior to application of any compounds. AUT1 (Autifony Therapeutics Ltd) was dissolved in a vehicle of sterile water containing 0.5% w/v HPMC 15M and 0.5% w/v Tween80™. 20 % w/w Captisol was included to increase solubilisation of AUT1 and was always included in the vehicle. Data presented are the outcome of 3 separate trials of different paradigms and doses of AUT1 (Autifony Therapeutics Ltd); **1)** 3 month (n=4) vs 20 month (n=4) using 60 mg/kg AUT1, **2)** 3 month (n=8) using 30 mg/kg AUT1 **3)** 18 month (n=8) using 30 mg/kg AUT1. In trial 1, mice received six sequential sessions, the first three being vehicle and the last three being the compound. In trials 2 and 3 mice were randomly assigned to two groups to alternate the sessions in which compound or vehicle was administered. Doses of AUT1 were administered at a volume of 0.1 ml/20 g intraperitoneally that corresponded to the correct dose. In each condition a 1 ml/20 g subcutaneous injection of saline was also administered to ensure voiding. Void samples and

frequency were recorded at 15/30 minute intervals over a 3 hour period on filter paper placed below the cages, using the void stained paper method (VSOP) (Sugino et al., 2008).

4.2.2 Analysis

Void stained filter papers were viewed with UV head goggles. The perimeter of each void was traced with a marker pen, imaged from a set height with a phone camera and measured using Image J calibrated to a ruler image under the same conditions. The area of a void stain was calibrated to the area produced by known volumes of mouse urine to derive the volume of a void. The observer was blinded in the sense that the only information present at the time of analysis was the session date and mouse ID.

For 3 month mice and 18 month mice, the level of sedation was assessed every 15 minutes with a score assigned based on the Sedation Rating Scale, a score of 0 reflecting asleep, 1- heavy sedation, 2- moderate sedation, 3 -mild sedation, 4-awake, inactive and 5-awake, active (Chuck et al., 2006). Due to the unblinded nature of the sedation assessment, a second blinded observer was used during a session to determine an inter-observer correlation of 0.76. For a measurement of locomotor activity, a widefield action camera was placed above the metabolic cages, recording for the entirety of the 3 hour session at 30 fps. Mice were tracked using Idtracker software (Perez-Escudero et al., 2014) and results are reported as the percentage of frames in which a mouse had displaced from the previous frame.

4.2.3 Statistics

All statistics were performed in R statistical software. All data were assessed for normality and homogeneity of variance using Shapiro Wilks and Levenes test, respectively. Trial 1 was a comparison between two conditions for multiple variables (multi-variate). Since some variables were normal and others not, single t-tests or Wilcox tests were carried out on select variables of interest and p-values were corrected for multiple comparisons using the Holm method. Trial 2 was a comparison between three conditions, thus repeated measures ANOVAs and Friedman tests were carried out on select parametric and no-parametric variables. Pairwise comparisons were made using either t-tests or Wilcox tests that were automatically corrected in the R package.

4.3 Results

4.3.1 Does specific modulation of Kv3 affect micturition?

Having established Kv3 immunoreactivity in excitatory and inhibitory inputs onto putative motoneurones in Chapter 1, we used a novel Kv3 modulator, AUT1 (Autifony Ltd) to test the hypothesis that Kv3 channels are functionally involved and important in the micturition reflex. 3 month mice were given either vehicle or the AUT1 compound and void frequency, total volume and volume/void were measured over a 3 hour period post-injection.

A formulation of 60 mg/kg of AUT1 (trial 1, see Materials and Methods) was made. This formulation eliminated void frequency, total volume and volume/void in the first hour in 3 month mice (n=4), with no significant effect in the following hours (**Fig. 4.1 A i, ii, iii)**. Over the 3 hours, a significant reduction in void frequency (50%, $p<0.001$) and total volume (37%, $p<0.001$) was recorded, with no significant difference in the volume/void between vehicle and AUT1 **(Fig. 4.1 B, ii, iii)**. In these experiments, mice were subjected to a subcutaneous injection of saline to normalise and ensure voiding. It is of note that this paradigm produced significant increases in void frequency and total volume over the 3 hour period with respect to values obtained from the acclimatisation period **(Fig. 4.1 D, E)**. Assessing the average time taken to a first void highlighted a significant delay of approximately 1.5 hours with the AUT1 compound ($p<0.05$) **(Fig. 4.1 C)**. In summary, 60 mg/kg of AUT1 acutely decreased bladder output.

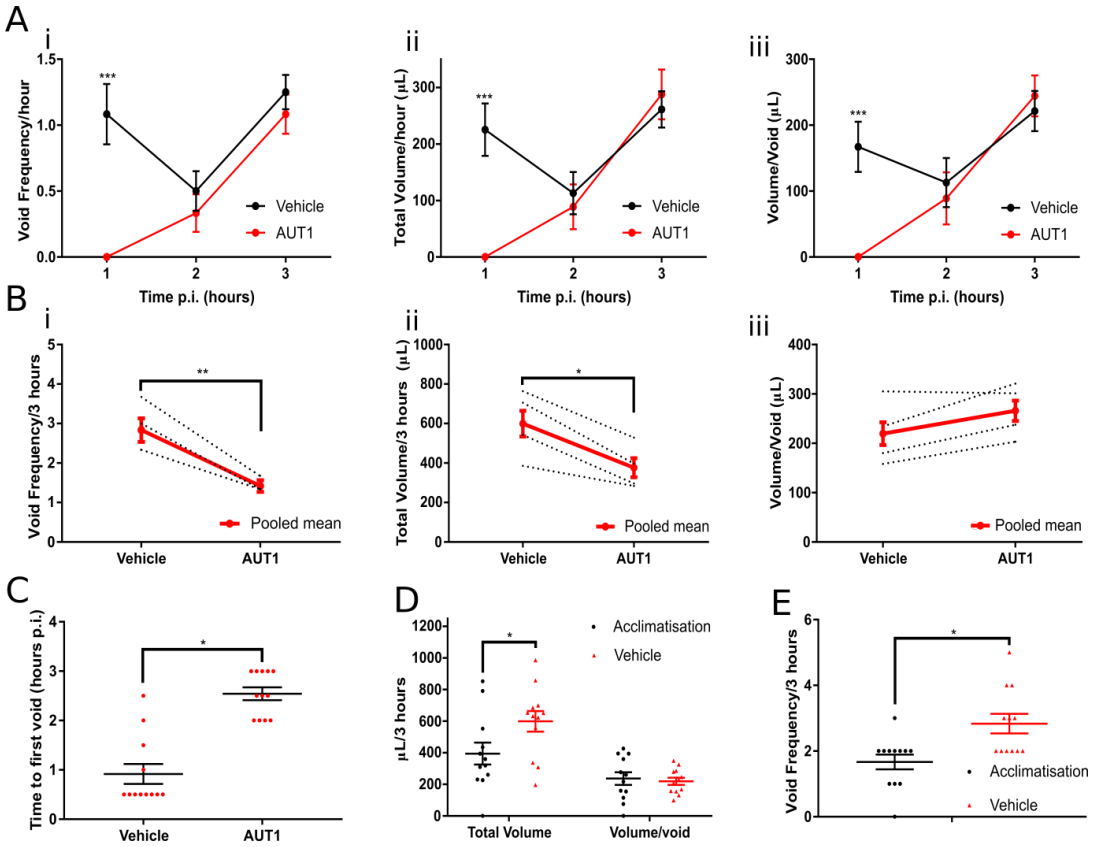


Figure 4.1. AUT1 (60mg/kg) transiently eliminated micturition in adult mice.

A) *AUT1 (60mg/kg) compared against vehicle at 1, 2 and 3 hours post injection (p.i.) in measures of bladder output; void frequency* ***(i)****, total volume* ***(ii)*** *and volume/ void (iii). C57bl6 (n=4) were subjected to each condition on three separate occasion. AUT1 (60mg/kg) compared against vehicle over the 3 hour session for bladder output; void frequency* ***(i)****, total volume* ***(ii)*** *and volume/ void* ***(iii)****-dashed lines reflect the mean individual response of individual mice and the red line is the pooled mean.* ***C)*** *Comparison of time to first void.* ***D+E)*** *Comparison of acclimatisation and administration of vehicle and s.c. saline. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

4.3.2 The reduction in bladder output is dose-dependent

Whilst assays of AUT1 against various other biological targets have demonstrated little cross reactivity (Rosato-Siri *et al.*, 2015) we cannot rule out an off-target site of action. To reduce the possibility that this pronounced reduction in micturition was due to an off target effect, we reduced the formulation concentration of AUT1 to 30 mg/kg in trial 2 (see Materials and Methods, n=8). At this lower dose we observed a significant reduction in void frequency (1.54 ± 0.19 to 0.88 ± 0.18 voids), total volume (315 ± 44.7 µl to 118 ± 42.2 µl) and volume/void (253 ± 15.5 µl to 118 ± 24 µl) in the first hour ($p<0.01$). This was followed by no change in the second hour but a significant increase in void frequency (0.67 ± 0.12 to 1.46 ± 0.15 voids) and total volume (191 ± 33 µl to 351 ± 35 µl) but not volume/void (290 ± 13 µl to 219 ± 17 µl) in the third hour compared to vehicle **(Fig. 4.2 A i, ii, iii)**. Over the three hour period, void frequency and total volume were not significantly different from vehicle, whereas volume/void was significantly reduced by 24% ($p<0.01$) from vehicle **(Fig. 4.2 B i, ii, iii)**. The time to first void was not significantly different between control and AUT1 **(Fig. 4.2 D)**. Experimenter observations during the higher 60 mg/kg dose trial had suggested that there could be a sedative effect associated with the novel compound. As a result, mice were assessed and scored every 15 minutes according to a sedation rating scale, with a score of 5 reflecting 'awake and active', and a score of 0 reflecting 'asleep' **(see Materials and Methods**). This sedation rating score was significantly lower during treatment with AUT1 in the first hour (3.8) compared to vehicle (4.7, $p<0.001$), reflecting a change from 'awake, active' to 'awake, inactive' **(Fig. 4.2 C).** There were no significant changes in the sedation rating between the two groups at

the second and third hour. These results indicated that both 30 mg/kg and 60 mg/kg of AUT1 acutely reduced bladder output and there is a potential confounding effect on sedation and activity.

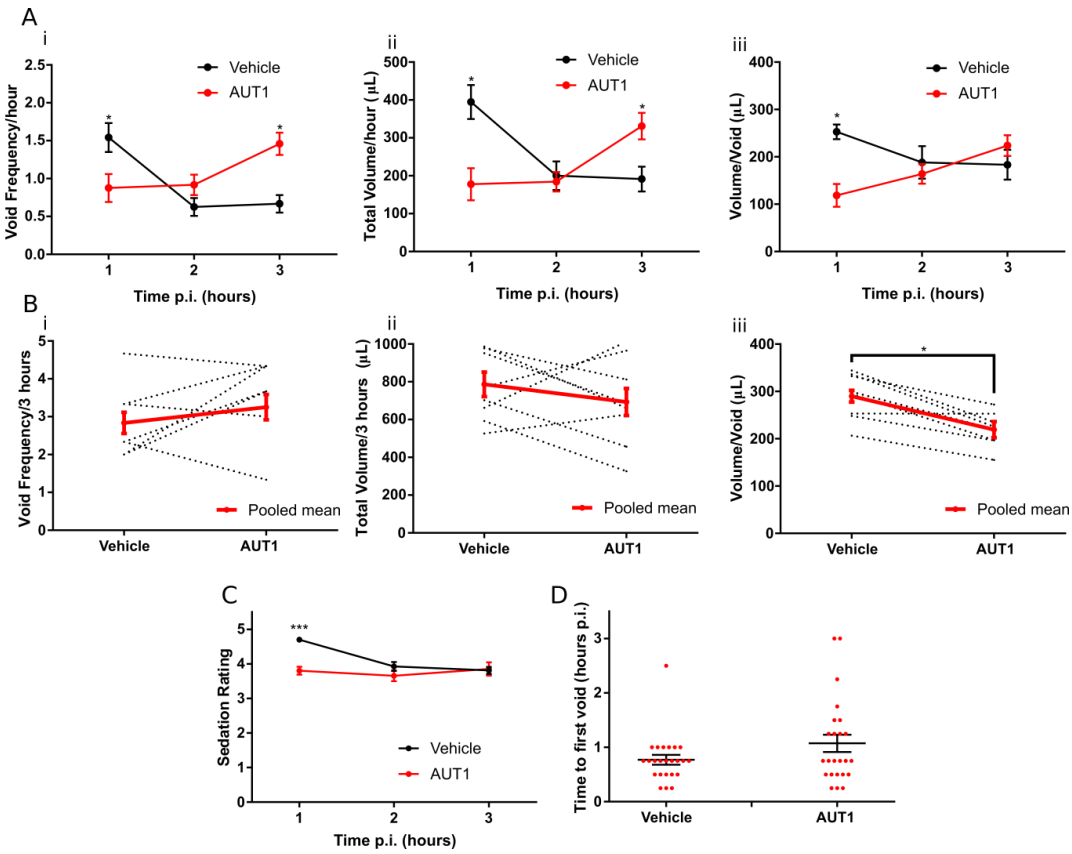


Figure 4.2. AUT1 (30 mg/kg) significantly reduced micturition and activity compared to vehicle treatment.

A) *A lower dose of AUT1 (30 mg/kg) compared to vehicle at 1, 2 and 3 hours post injection (p.i.) in measures of bladder output; void frequency* ***(i)****, total volume* ***(ii)*** *and volume/ void* ***(iii). C57bl6 (n=8) were subjected to each condition on three separate occasions B)*** *AUT1 (30mg/kg) vs vehicle over the 3 hour session for bladder output; void frequency* ***(i)****, total volume* ***(ii)*** *and volume/ void* ***(iii)****-dashed lines reflect the mean responses of individual mice and the red line is the pooled mean.* ***C)*** *Sedation rating scores for both groups.* ***D)*** *A comparison of time to first void for both groups. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

4.3.3 Does the response to modulation differ with age?

We previously explored age-related changes in immunoreactivity of Kv3 channels in the vicinity of putative bladder motoneurones finding significant decreases with age. We therefore postulated that modulation of bladder output by AUT1 could be susceptible to changes in age too and that the demonstrated recovery effect of AUT1 in vitro (Rosato-Siri et al., 2015) could have clinical relevance for age-related incontinence. 20 month mice subjected to the same protocol and a 60 mg/kg formulation responded similarly to 3 month mice, however with some differences in response to the experimental paradigm as discussed below. In the aged mice, AUT1 eliminated void frequency, total volume and volume/void in the first hour, however, this was against the backdrop of a very low rate and volume of bladder output in the vehicle condition in the first hour **(Fig. 4.3 A i, ii, iii)**. A significant reduction, 77%, 89% and 85% in void frequency, total volume and volume/void ($p<0.05$), respectively, compared to vehicle was observed over the 3 hour period **(Fig. 4.3 Bi, ii, iii)**. No significant difference in time to first void was observed **(Fig. 4.3 C)**.

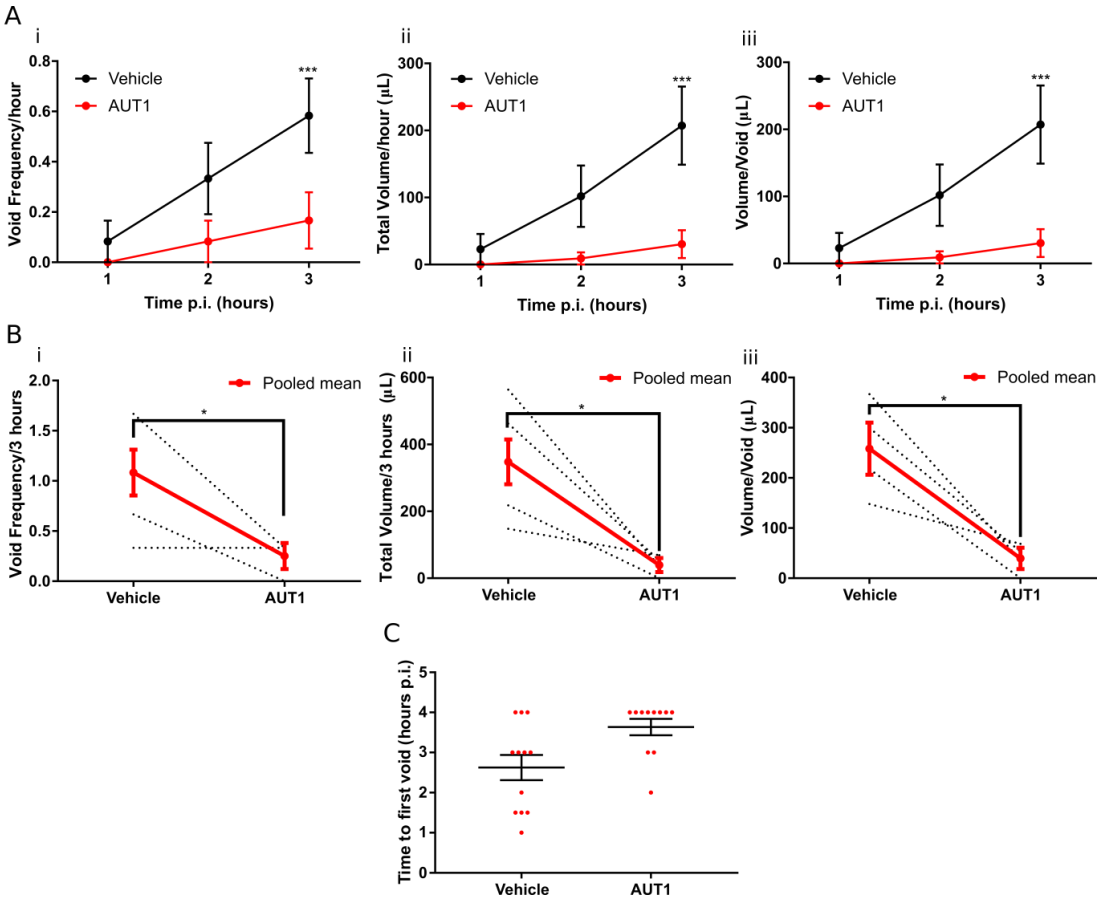


Figure 4.3. Aged mice respond differently to AUT1 (60 mg/kg) and the experimental paradigm.

A) *In 20 month mice (n=4), AUT1 (60 mg/kg+captisol) compared against vehicle at 1, 2 and 3 hours post injection (p.i.) in measures of bladder output; void frequency* ***(i)****, total volume* ***(ii)*** *and volume/ void* ***(iii). B)*** *Overall change from vehicle.* ***C)*** *A comparison of time to first void. Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

4.3.4 20 month mice do not display an incontinent phenotype

In a comparison of young and aged mice during acclimatisation, we found that 20 month mice voided at the same rate but at larger volumes when compared with their 3 month counterparts ($p<0.05$)**(Fig. 4.4 B i, ii)**. Subcutaneous injection of saline in the vehicle condition produced a sub-normal phenotype (i.e. lower bladder output than in acclimatisation) when compared with void frequency, total volume and volume/void ($p<0.05$) from the acclimatisation phase, and the results should be interpreted in consideration of this **(Fig. 4.4 A i, ii)**.

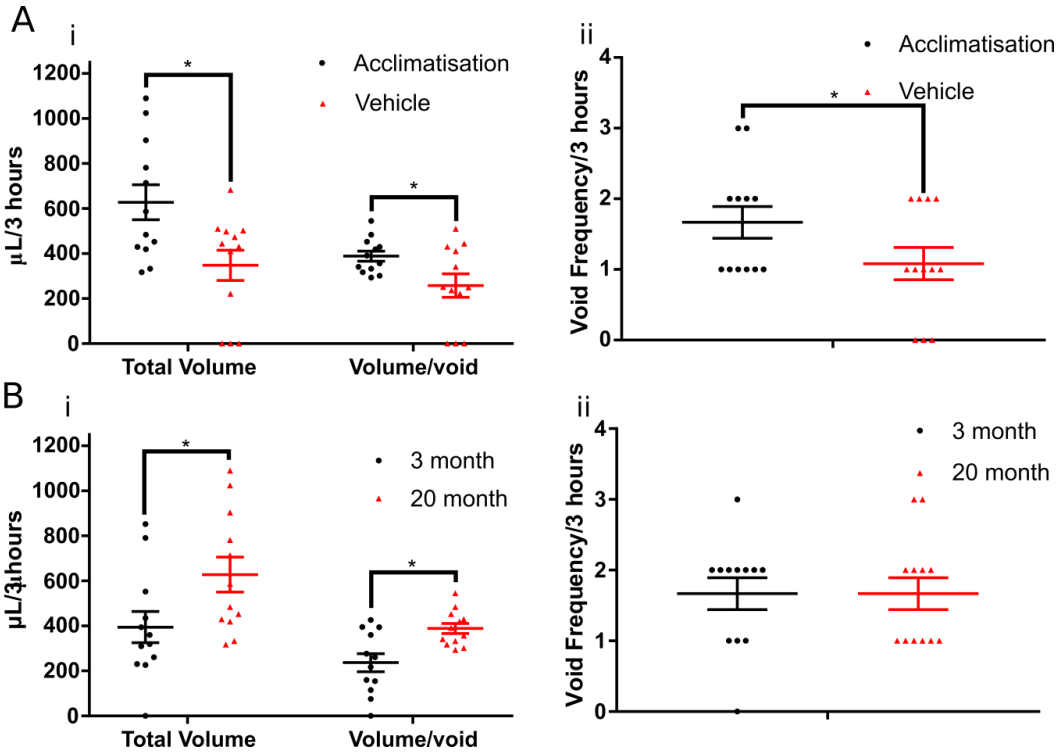


Figure 4.4. Bladder output in 20 month mice compared to acclimatisation and 3 month mice.

A) *A comparison between acclimatisation and vehicle for void volume (i) and frequency (ii).* ***B)*** *A comparison between 3 month and 20 month mice for void volume (i) and frequency (ii). Data represents individual sessions from C57bl6 mice (n=4 for each condition, each mouse was subjected to each condition on three occasions.) Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

4.3.5 Sedation Rating Scores were validated by video tracking

In a separate experiment with 18 month old mice, at the lower 30 mg/kg dose, we found little significant difference in bladder output (not shown) but a significant reduction in the sedation rating score, that was sustained across the 3 hour session, again from 'awake, active' to 'awake, inactive' ($p<0.001$) **(Fig. 4.5 A)**. As this appeared to be a reduction in activity, a camera was placed above the cages and locomotor activity was recorded for the entire session. Analysis of these data obtained in this method supported the significant trend observed in the sedation score, with significant reductions in the percentage of frames in which a mouse was active being significantly reduced in the AUT1 condition compared to vehicle control group, across all three time points ($p<0.05$) **(Fig.4.5 B)**.

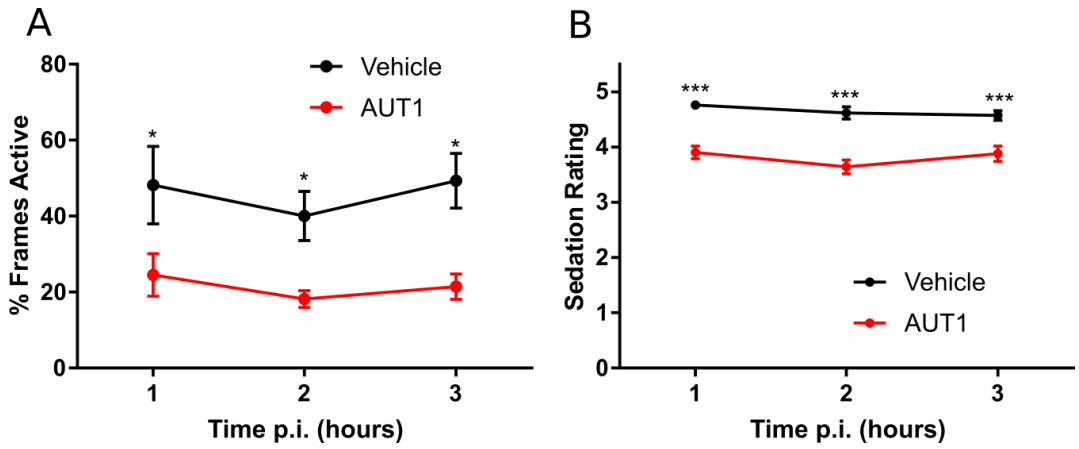


Figure 4.5. 30 mg/kg reduced sedation rating score and locomotor activity in 18 month mice.

A) *The percentage of video frames in which the mice were active.* ***B)*** *Sedation rating scores for both groups.* ***N=8, three sessions per condition.*** *Error bars are SEM and p<0.05=*, p<0.01=**, p<0.001=***.*

4.4 Discussion

4.4.1 Novel Kv3 modulation reduces bladder output

We found that novel modulation of Kv3 channels with AUT1 reduced bladder output acutely compared to a control group. This acute effect occurred in a dose dependent manner with acute elimination of micturition at the highest dose and a significant reduction at the lowest.

So where could AUT1 be working? The immediate candidates in the micturition reflex are bladder afferents, pontine micturition centre (PMC) neurones, nodes of Ranvier, spinal interneurones, synapses, the neuromuscular junction of the external urethral sphincter (EUS) and the bladder detrusor muscle itself. At the time of writing, Kv3 channels are not known to be expressed in central afferent terminals at this spinal level, however Kv3 channels are expressed in afferent terminals in other brain regions (Ishikawa et al., 2003; Dallas et al., 2005; Dallas et al., 2008) therefore exploration of Kv3 subunits in spinal afferent terminals is a worthy pursuit. In dorsal root ganglia (DRG), where peripheral sensory input is integrated, only Kv3.4 subunits are expressed. As Chapter 2 demonstrated, AUT1 reduced homomeric Kv3.4 conductance and in DRG neurones, Kv3.4 accounts for around 40-70% of total repolarisation (Ritter *et al.*, 2015). Therefore, reduction of Kv3.4 current would result in reduced repolarisation efficiency, hyperexcitability and afferent sensitisation. In the context of bladder circuitry this would mean that afferent activity that predominates in the guarding reflex during continence is intensified and would strongly promote sympathetic outflow, detrusor inhibition and continence. This would fit the data

presented here. The PMC (also known as Barrington's nucleus) is a collection of neuronal cell bodies that receive afferent input from the bladder and govern the switch from continence to voiding. Using the Allen Brain In Situ Hybridisation atlas, we find no expression of RNA for Kv3 subunits in this nucleus. Interestingly, Kv3 subunits Kv3.3 and Kv3.4 are expressed in the neuromuscular junctions of transversus abdominus, lumbrical and flexor digitorum brevis muscles (Brooke et al., 2004), however whether they are expressed in the NMJ of the external urethral sphincter is unknown. If we hypothesise that they are expressed here then we can postulate that AUT1 acts to inhibit Kv3.4 channels and weakly potentiate Kv3.3 channels (AUT1 is about ~3 fold less potent at Kv3.3 channels than Kv3.1b channels (Rosato-Siri et al., 2015). In doing so, one would expect the effect at Kv3.4 channels to dominate the synaptic response to AUT1, reducing repolarisation efficiency, broadening the NMJ action potential and increasing neurotransmitter release to the EUS to strongly promote continence. Whilst the detrusor muscle of the bladder does contain Kv currents, these are not consistent with the properties of Kv3 channels, rendering this muscle as an unlikely site of action (Thorneloe and Nelson, 2003; Petkov, 2011).

Finally, left to consider is action at spinal interneurones and nodes of Ranvier, both likely to be Kv3-positive (Deuchars et al., 2001; Brooke et al., 2002; Devaux et al., 2003; Nowak et al., 2011), and synaptic inputs, some of which we have established within the micturition reflex as Kv3-positive.

Hypothetically what would we expect the results of AUT1 modulation at the interneurone, node or the synapse to be, and how would this relate to the

pronounced effect on micturition that we have observed? In parvalbumin-positive TEA-sensitive cortical interneurones, 10 μM of AUT1 had no effect on firing frequency (Rosato-Siri et al., 2015). The maximal firing rate of these interneurones was 168 Hz but perhaps AUT1 has more of a pronounced effect in faster-firing neurones. Interestingly, the work of Brown et al. (2016) found that 10μM AUT1 reduced the number of action potentials in MNTB neurones stimulated at 200-600Hz and that sub-threshold responses were increased in amplitude. This would suggest that the effect of AUT1 is frequency dependent. But what would these results indicate for Kv3-positive spinal interneurones that fire at rates less than 100Hz (Deuchars et al., 2001)? Interestingly, results in Chapter 2 demonstrated that TEA-sensitive spinal interneurones in AUT1 fired at lower frequencies with more regularity and with larger and longer afterhyperpolarisations. In the context of bladder circuitry this would mean that interneuronal excitability is reduced and potentially against a backdrop of sensitised afferent inputs. Whilst GABAergic interneurones antecedent to autonomic neurones have been described, we can't exclude the possibility of TEA sensitive glutamatergic interneurones being involved in the micturition reflex. Therefore due to this ambiguity, at the time of writing it is unclear what effect this would have on bladder motoneurone output and the bladder reflex. It would be expected that Kv3-positive nodes of Ranvier described by Devaux *et al.*, (2003) would experience a similar suppression of excitability as that recorded at the cell soma.

At the synapse involved in the bladder circuitry and, in regards to the action potential itself, what shape would we expect to observe with incubation of

AUT1? Firstly, AP spikes originating from the axon initial segment (AIS) and those entering a pre-synaptic site are underlined by the expression of a variety of voltage-gated ion channels and can therefore be very different (Hoppa *et al.*, 2014). In addition, studies of AUT1 on neuronal firing highlight only changes to the shape of spikes from the AIS thus the effect on pre-synaptic AP waveforms is unclear. However, based on these studies and the results in Chapter 2, two main hypotheses of how AUT1 modifies the spike waveform exist. Briefly, the first goes like this; the shift in activation curve and open probability of Kv3.1 and Kv3.2 channels to more negative potentials by AUT1 in recombinant cell lines (Rosato-Siri et al., 2015; Brown et al., 2016) could indicate that Kv3-dependent currents are activated earlier in an action potential of a native neurone. This earlier activation combined with the potentiated current observed in recombinant Kv3 channels could suggest a greater repolarisation efficiency and a waveform with a lower amplitude and the shift to a more negative activation potential could potentially lead to a shift in the threshold required for action potential initiation, as found by Brown *et al.*, (2016). In these neurones, AUT1 appeared to prevent action potential initiation by way of an increased threshold for firing essentially suppressing excitability. The second hypothesis proposed in Chapter 2, also posits suppression of firing but instead by slowed deactivation of Kv3 channels that prolongs the afterhyperpolarisation (AHP) phase therefore only modifying the AHP shape, which is supported by data in this thesis. Essentially at the level of the synapse, hypothesis 1 might make it harder for a spike to fire and when it does with a smaller amplitude, whereas hypothesis 2 would restrict how many presynaptic spikes could fire in a given space of time. Both predict a suppression of the frequency of pre-synaptic

excitability and as Kv3 channels are localised to both excitatory and inhibitory synapses one would expect a general reduction in the rate of synaptic transmission onto bladder motoneurones.

4.4.2 A sedative effect?

The effect of AUT1 on bladder output is clearly accompanied by a reduction in activity or a sedative effect and whether this confounds data on bladder output is a concern but unclear at the time of writing. The effect appears to be more pronounced than that of the effect on bladder output, either indicating that it precedes the effect on the bladder or that the circuitry/mechanism underlying it is more sensitive i.e. containing more Kv3 targets than bladder circuitry. The effect of treatment with AUT1 on activity was also confirmed by Parek et al. 2018 using infra-red photobeams to detect locomotor activity. The reduction in activity increased with the dosage and was still present at high doses even in Kv3.1 KO and Kv3.2 KO mice indicating that these subunits contributed partially but not fully to the effect. Hypothetically, Kv3.4 inhibition by AUT1 at NMJs containing Kv3 channels could increase neurotransmission and may provide a tonic level of muscular contraction that also may contribute to a reduction in activity observed here. The effect of AUT1 on Kv3.4 subunits was unknown at the time of Parek et al (2018) but must be considered in light of this thesis. It also necessitates investigation into the effect of AUT1 on Kv3.4 subunits at the neuromuscular junction. Furthermore, because this effect is still present to a degree in Kv3 subunit knockouts it is likely that there is activity at a secondary site, within the central nervous system or within the periphery. Future

experiments using more specific modulators may remove the confounding effect of sedation and any other off target effects on bladder output.

4.4.3 Therapeutic relevance

Finally, in proposing therapeutic use in a condition such as incontinence, we have to address whether the mouse model used here recapitulates incontinence associated with ageing in humans. What we find is that although the AUT1 compound appears to have a more sustained reduction in bladder output, the aged mice themselves do not present with an incontinent phenotype, instead presenting one in which voids of greater volume are passed, most likely reflective of a larger bladder capacity. However, that this effect is significant and longer lasting in aged mice, could still have clinical relevance and should be explored in a model of incontinence and advanced aged mice where incontinence does develop.

Chapter 5

5 General Discussion

5.1 Summary

Data presented here focuses on the role of Kv3 channels in the murine spinal cord with a focus on lumbo-sacral spinal circuitry. Immunohistochemical analysis identified Kv3 subunits Kv3.1b and Kv3.3 in both excitatory and inhibitory synapses closely apposed to putative bladder motoneurones in the lumbosacral murine spinal cord. Interestingly, this expression of Kv3 subunits was reduced in aged mice. Stimulation of descending tracts evoked EPSPs in neurones recorded in the vicinity of the anatomical location of bladder autonomic motoneurones. A proportion of these EPSPs were sensitive to 0.5 mM TEA suggesting that Kv3 subunits play a functional role in spinal synapses in the lumbo-sacral spinal cord. Application of a Kv3 modulator in HEK cells transfected with Kv3.1b and Kv3.4a cDNA shifted the activation curve of currents in Kv3.1b cells, increased the rate of inactivation of currents in Kv3.4a cells and decreased the rate of deactivation in both. Co-transfection produced putative heteromeric currents also negatively shifted by application of AUT1 suggesting that the effect of AUT1 isn't specific to homomers. In vitro pre-incubation of lumbo-sacral spinal cord slices in AUT1 suggested that AUT1 suppressed neuronal firing by increasing the afterhyperpolarisation, this being contrary to previous hypotheses. Interestingly, suppressed firing of interneurones at the level of bladder circuitry may affect bladder output. To investigate this AUT1 was administered to mice and bladder output monitored. High doses (60 mg/kg) of AUT1 eliminated bladder output whereas lower doses

(30 mg/kg) attenuated output. These results however were accompanied by a sedative effect or general reduction in activity, the underlying mechanism and site of action being unknown. This chapter will focus on the significance of the results, limitations associated with the presented results and will define important next steps in elucidating the role of Kv3 channels in the spinal cord.

5.2 Stoichiometry in co-transfection experiments

Co-transfection of Kv3.1b and Kv3.4a cDNA led to distinct putative heteromeric currents. However, many of the features of these currents were reproduced by summation of a large proportion of Kv3.1b currents with a small proportion of Kv3.4a currents leading to some uncertainty of the true nature of the heteromeric currents. Additionally, co-transfection offers little control over stoichiometry of Kv3 subunits within a channel. Ideally, using plasmids where cDNA for ideal stoichiometry of subunits are concatenated together increases the local concentration of these subunits and the likelihood of them tetramerising to form defined heteromers (Sack, Shamotienko and Dolly, 2008). This would allow for greater certainty and control of defining heteromeric Kv3 channels for use in pharmacological screens such as performed here.

5.3 Independent comparisons within a heterogeneous neuronal population

Interneurones within lamina VII of lumbo-sacral spinal cord are poorly characterised compared to other regions and levels of the spinal cord, such as laminae I-III in the superficial dorsal horn of the spinal cord (Todd, 2010, 2017), and may represent many distinct cell types based on morphology, electrophysiology and neurotransmitter content as described in other brain

regions (Markram *et al.*, 2015). Therefore independent analysis of a small potential heterogeneous sample of neurones within this region may limit comparison in conditions where slices are pre incubated in either control or AUT1 aCSF, meaning as large a sample size as possible would be a significant improvement. There exists an obvious disparity between the onset of AUT1 in cell lines and in *in vivo* compared with in slices, however the underlying reason is unknown. 90 minutes after intra-peritoneal injection of 60 mg/kg AUT1, free brain concentrations are in the range of between 0.1 to 2.1 μM (Parek et al, 2018) which should modulate Kv3.1 currents (EC50 is 5 μM) by between 0 and 20 % (Rosato-Siri et al, 2015). These concentrations are likely to be larger earlier but indicate two things; AUT1 may be cleared or metabolised rapidly perhaps explaining the acute of effect on bladder output observed here, or we may be observing a peripheral site of action as the free brain concentrations are possibly too low to significantly modulate Kv3 channels. An interesting complement to these experiments would be the use of dynamic clamp. The ability to add or subtract control and AUT1-modified Kv3 conductances obtained from heterologous expression systems, could be used to directly test the effect of an AUT1-modified current on neuronal firing within the same neurone.

5.4 The micturition paradigm

There exists a clear limitation in the micturition paradigm. Intra-peritoneal application of either the compound or vehicle often resulted in the mice voiding during handling. Sub-cutaneous injections of saline were thus given at the same time to normalise loading and ensure voiding of the bladder based on animal

weight. This appeared to have a different effect on young and aged mice where young mice voided more compared to acclimatisation and aged mice voided less, and this probably reflected a much larger volume of the bladder in the aged mice. Decoupling handling of the mice from administration by the use of osmotic mini-pumps that release compound over a set period of time would resolve more natural behaviour.

5.5 Action potentials recorded at the soma

Preliminary analysis of recovered neuronal morphologies, although not formally quantified, suggested that in some neurones the axon originates from a dendritic branch. This could place the axonal initial segment (AIS) some distance from the soma where patch clamp recordings were performed. Therefore an important consideration is that the action potentials recorded more accurately reflect the membrane potential changes at the soma rather than the action potential initiated and propagated along an axon (Bean, 2007). Immunohistochemical analysis shows Kv3 channels expressed in cell bodies but it is currently unknown whether Kv3 channels are expressed at the AIS of spinal neurones. Therefore there is an important distinction and separation to make between the effect of Kv3 channels expressed in the AIS and those expressed in the soma on the shape of the recorded action potential. Thus, in the analysis of the effect of AUT1, the reported shape of the AP waveform recorded at the soma may be a less reliable indicator of effect compared to the reported firing properties such as frequency, interspike intervals and variation (Bean, 2007). Ankyrin G is a marker of the AIS (Duflocq et al, 2011) and co-localises with Kv3.1b channels in the axonal membrane. In combination with a somato-

dendritic marker such as MAP2 (Duflocq et al, 2011) one could investigate the expression of Kv3 subunits at the AIS and the influence on of action potential shape at initiation.

5.6 Electrophysiological recordings at non-physiological temperature

All electrophysiological data presented in this thesis was recorded at room temperature, and thus an obvious limitation exists in the interpretation of native channel and native neuronal function. The biological effect of temperature on processes such as gating is determined by a 10 degree temperature coefficient (Hille, 2001). A rise in temperature leads to an exponential rise in the rate of gating, meaning that channel activation and deactivation recorded at room temperature can assumed to be significantly faster in a more physiological situation. Indeed, recordings of Kv3.1 channels at physiological temperatures induced a faster rate of inactivation (Oliver et al, 2017). Whilst the conductance is relatively insensitive to temperature (Hille, 2001), the faster rates of activation and inactivation mean that the conductances during an action potential and thus the action potential itself are more rapid. Therefore, the data presented here must be interpreted with this in mind.

5.7 Recovery of Kv3.4 channels from inactivation

For cells transfected with Kv3.4 cDNA, currents rapidly inactivated. However, experiments to gauge the time course of recovery from inactivation weren't performed and a recovery time of 4 seconds at -70 mV was used. Whilst this

seemed sufficient in control conditions, it may be more important especially after application of AUT1 where the voltage dependence of inactivation was negatively shifted. A true reflection of the AUT1 effect on Kv3.4a channels would be achieved with a longer recovery such as 10 seconds, at more hyperpolarised potentials (Hartmann *et al.*, 2018). This group varied interpulse intervals in order to plot the time interval required after inactivation to achieve the same current response as the initial test. A second consideration regarding Kv3.4 expression was intracellular dialysis during whole cell recordings, as it was observed that Kv3.4 currents lost inactivation during long recordings with long depolarisations. This could be hypothesised to be due to PIP2 depletion and PKC activation, therefore an alternative approach would be the use of the perforated patch technique to prevent dialysis. This benefit would have to be weighed against the increased access resistance associated with this technique and the size of the currents recorded.

5.8 Future work

There is large scope for future work based on the preliminary findings in this thesis, from exploring the effect of AUT1 in incontinence models to fully understanding the Kv3 contribution to excitability in spinal neurones.

5.8.1 In cell lines

This thesis argues that the potentiation of Kv3.1b and reduced conductance of Kv3.4a is due to slowed deactivation by AUT1 that favours channel opening and, in Kv3.4a channels, open-state inactivation (Fineberg, Ritter and Covarrubias, 2012). To test this hypothesis it would be interesting to either engineer a non-inactivating Kv3.4a channel (Kv3.4IR) or to include a protease enzyme such as

papain in the intracellular solution to remove the inactivation structure (Hille, 2001). In this scenario, if AUT1 still induces a slower deactivation and a reduced conductance then it must be inactivation independent however if potentiation, like in Kv3.1b, is observed then it is inactivation-dependent and at a site similar to that of Kv3.1b. In addition, to thoroughly address the slower deactivation of Kv3.4a, briefer depolarising pulses could be used to limit Kv3.4 inactivation, where it was only possible to compare one short depolarising pulse in the dataset obtained during this thesis.

5.8.2 In the spinal cord

The widespread distribution of Kv3 subunit immunoreactivity within the spinal cord appeared to be localised to synaptic, axonal and somatic structures. Future work in the lumbo-sacral spinal cord should focus on the role of Kv3 channels within these structures. Much of the literature in the spinal cord has focused on Kv3.1b channels in the firing of interneurones and localisation to nodes of Ranvier (Deuchars *et al.*, 2001; Brooke *et al.*, 2002; Devaux *et al.*, 2003) and the functional role that inactivating Kv3 subunits, Kv3.3 and Kv3.4, play in these structures has been understudied. Additionally, the role of Kv3.3 channels extends beyond K^+ conductance to interaction with the actin cytoskeleton via the C-terminus (Zhang et al, 2016). Investigation into other binding interactions of Kv3.3 channels are in their infancy, but may provide promising insights into the role of Kv3 channels within a cell. While the role of Kv3 channels in spinal synapses has been tested here using TEA, further separation of Kv3 subunit contributions to synaptic transmission could be made by using Kv3.4 blocker, blood-depressing-substance (BDS) (Dallas et al, 2008). Additionally, where

specific blocking agents do not exist for certain subunits, subunit-specific primary antibodies could be and have been used as an alternative (Dallas et al, 2008). Importantly, this thesis has focussed on heteromers in an effort to replicate native physiology, however, while Kv3 heteromers are likely to occur in the spinal cord, definitive proof of their existence is needed. Further work should combine directly conjugated antibodies with super-resolution microscopy to visualise the spatial co-localisation of different Kv3 subunits. Additionally, co-immunoprecipitation could be used to determine physical interaction between subunits and the types of heteromers formed. The fundamental question unanswered by this thesis is what do the Kv3 or high-voltage potassium currents look like in spinal interneurones. Therefore, voltage clamp of these neurones using standard step protocols and AP waveform protocols is essential to understand the role and onset of these currents during firing. Due to the prominent expression of Kv3.1 and Kv3.3 subunits in the spinal cord, the currents expected would likely be those of a high-voltage-activated delayed rectifier, a high-voltage-activated inactivating current and possibly a heteromeric combination of both. Additionally, voltage clamp would be a more appropriate technique for analysis of the effect of TEA on synaptic transmission. This is because application of TEA would also block potassium channels on the postsynaptic membrane, decreasing conductance and increasing postsynaptic membrane resistance. An increase in membrane resistance would increase the time constant of the cell ($\tau=RC$) and voltage response to a synaptic current ($V=IR$), confounding both the amplitude and duration of postsynaptic potentials recorded in current clamp. Solely measuring the postsynaptic current instead of potentials in voltage clamp would

circumvent this confounding effect on the passive properties of the neurone. Furthermore there are important considerations for future work using TEA to extract Kv3 mediated effects and currents. For example, TEA in the millimolar range can block other potassium channels. This can affect the fidelity of the voltage clamp in neurones and confound current subtraction. In addition other compounds such as DTX should be used to sequentially block these other TEA-sensitive channels like Kv1, to specifically extract the effect of Kv3 on synaptic transmission (Johnston et al, 2010). As eluded to earlier, the characterisation of lamina VII interneurones in the lumb-sacral spinal cord is lacking. There is therefore scope for characterising the electrophysiological, morphological and neurochemical phenotypes of cells in this region that would improve our ability to interpret circuit level and behavioural level modulations. Detailed characterisation also permits more accurate neuronal models into which normal Kv3 conductances and AUT1-modified conductances could be inserted to simulate and make predictions about the AUT1 effect on native neurones as a pre-requisite to direct experimentation.

5.8.3 In vivo

Intra-peritoneal administration of AUT1 evidently eliminated bladder output at high doses and reduced output at lower doses. A part of the impact of this project was a focus on Kv3 channels in ageing and bladder output in ageing. Importantly, however, in mice aged 20 months no signs of incontinence were observed. Other studies suggest that increases in void frequency in mice occur at more advanced ages (>28 months, Merican 2016, unpublished) so perhaps

future studies examining the therapeutic value of AUT1 in age-related incontinence should be carried out in very old mice (>28mths). Another avenue would be the use models of over-active bladder disorder which can be divided into being either of peripheral or central cause. Examples of peripheral models include inducing afferent hypersensitivity using chemicals such as capsaicin or acetic acid (Parsons and Drake, 2011). Interestingly, the spontaneously hypertensive rat (SHR), commonly used to study what is says on the tin, displays bladder hyperactivity (Parsons and Drake, 2011). This hyperactivity was underlined by changes in noradrenergic control of the micturition reflex, as intrathecal adrenergic block using doxazosin reduced micturition pressure (Persson *et al.*, 1998). This was thought to be because sacral interneurones containing adrenergic receptors receive noradrenergic input from the pons as part of the locus coeruleus noradrenergic system and in turn excited parasympathetic preganglionic motoneurones to induce contraction of the bladder (Yoshimura *et al.*, 1990; Persson *et al.*, 1998). In light of this and the central alterations, SHR may make a useful model in vivo and in vitro. While studies carried out here used female mice, bladder outlet obstruction models designed to replicate benign prostatic enlargement in humans can also be used to study incontinence (Parsons and Drake, 2011). Finally, for central effect models, transgenic knock out of nitric oxide synthase can be used as well as other disease models that recapitulate bladder dysfunction such as in multiple sclerosis and spinal cord injury (Parsons and Drake, 2011).

5.9 Conclusion

Kv3 channels are an important component in determining excitability of many cell types in the central nervous system, at the soma, axon and synapse. Of interest in the data presented in this thesis has been the role of Kv3 channels in relatively fast firing spinal interneurones and synapses apposed to putative bladder motoneurones. Evidence presented here suggested that Kv3 channels are expressed in both excitatory and inhibitory synapses, that selective modulation of Kv3 channels with AUT1 alters the firing of interneurones and that systemic administration alters bladder output. Fundamentally, this work highlights the physiological implications of pharmacologically targeting Kv3 channels and identifies an effect that may be of therapeutic value to conditions of overactive bladder.

List of References

Alle, H., Kubota, H. and Geiger, J. R. P. (2011) 'Sparse But Highly Efficient Kv3 Outpace BKCa Channels in Action Potential Repolarization at Hippocampal Mossy Fiber Boutons', *Journal of Neuroscience*, 31(22), pp. 8001–8012. doi: 10.1523/JNEUROSCI.0972-11.2011.

Araki, I. and de Groat, W. C. (1997) 'Developmental synaptic depression underlying reorganization of visceral reflex pathways in the spinal cord.', *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 17(21), pp. 8402–7. Available at: http://www.ncbi.nlm.nih.gov/pubmed/9334413.

Araki, I. and De Groat, W. C. (1996) 'Unitary excitatory synaptic currents in preganglionic neurons mediated by two distinct groups of interneurons in neonatal rat sacral parasympathetic nucleus', *J Neurophysiol*, 76(1), pp. 215–226. doi: 10.1152/jn.1996.76.1.215.

Baranauskas, G. *et al.* (2003) 'Kv3.4 subunits enhance the repolarizing efficiency of Kv3.1 channels in fast-spiking neurons', *Nat Neurosci*. 2003/02/20, 6(3), pp. 258–266. doi: 10.1038/nn1019.

Barry, J. *et al.* (2013) 'Activation of conventional kinesin motors in clusters by Shaw voltage-gated K+ channels', *J Cell Sci*. 2013/03/15, 126(Pt 9), pp. 2027–2041. doi: 10.1242/jcs.122234.

Bautista, W. and Nagy, J. I. (2014) 'Connexin36 in gap junctions forming electrical synapses between motoneurons in sexually dimorphic motor nuclei in spinal cord of rat and mouse', *European Journal of Neuroscience*, 39(5), pp. 771–787. doi: 10.1111/ejn.12439.

Bean, B. P. (2007) 'The action potential in mammalian central neurons', *Nature Reviews Neuroscience*, 8(6), pp. 451–465. doi: 10.1038/nrn2148.

Blok, B. F. M., Van Maarseveen, J. T. P. W. and Holstege, G. (1998) 'Electrical stimulation of the sacral dorsal gray commissure evokes relaxation of the external urethral sphincter in the cat', *Neuroscience Letters*, 249(1), pp. 68–70. doi: 10.1016/S0304-3940(98)00382-6.

Boda, E. *et al.* (2012) 'Brain expression of Kv3 subunits during development, adulthood and aging and in a murine model of Alzheimer's disease', *J Mol Neurosci*. 2011/09/14, 46(3), pp. 606–615. doi: 10.1007/s12031-011-9648-6.

Boddum, K. *et al.* (2017) 'Kv3.1/Kv3.2 channel positive modulators enable faster activating kinetics and increase firing frequency in fast-spiking GABAergic interneurons', *Neuropharmacology*, 118, pp. 102–112. doi: 10.1016/j.neuropharm.2017.02.024.

Boscia, F. *et al.* (2017) 'The expression and activity of KV3.4 channel subunits are precociously upregulated in astrocytes exposed to Aβ oligomers and in astrocytes of Alzheimer's disease Tg2576 mice', *Neurobiology of Aging*, 54, pp. 187–198. doi: 10.1016/j.neurobiolaging.2017.03.008.

Brew, H. M. and Forsythe, I. D. (1995) 'Two voltage-dependent K+ conductances with complementary functions in postsynaptic integration at a central auditory synapse.', *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 15(12), pp. 8011–8022.

Brooke, R. E. *et al.* (2002) 'Spinal cord interneurones labelled transneuronally from the adrenal gland by a GFP-herpes virus construct contain the potassium channel subunit Kv3.1b', *Auton Neurosci*. 2002/07/30, 98(1–2), pp. 45–50. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12144039.

Brooke, R. E., Atkinson, L., *et al.* (2004) 'Association of potassium channel Kv3.4 subunits with pre- and post-synaptic structures in brainstem and spinal cord', *Neuroscience*. 2004/06/23, 126(4), pp. 1001–1010. doi: 10.1016/j.neuroscience.2004.03.051.

Brooke, R. E., Moores, T. S., *et al.* (2004) 'Kv3 voltage-gated potassium channels regulate neurotransmitter release from mouse motor nerve terminals', *Eur J Neurosci*. 2004/12/22, 20(12), pp. 3313–3321. doi: 10.1111/j.1460-9568.2004.03730.x.

Brooke, R. E. *et al.* (2006) 'Immunohistochemical localisation of the voltage gated potassium ion channel subunit Kv3.3 in the rat medulla oblongata and thoracic spinal cord', *Brain Res*. 2006/01/13, 1070(1), pp. 101–115. doi: 10.1016/j.brainres.2005.10.102.

Brooke, R. E. *et al.* (2010) 'Kv3.3 immunoreactivity in the vestibular nuclear complex of the rat with focus on the medial vestibular nucleus: targeting of Kv3.3 neurones by terminals positive for vesicular glutamate transporter 1', *Brain Res*. 2010/05/18, 1345, pp. 45–58. doi: 10.1016/j.brainres.2010.05.020.

Brown, M. R. *et al.* (2016) 'Physiological modulators of Kv3.1 channels adjust firing patterns of auditory brain stem neurons', *J Neurophysiol*. 2016/04/08, 116(1), pp. 106–121. doi: 10.1152/jn.00174.2016.

Catterall, W. A. *et al.* (2007) 'Voltage-gated ion channels and gating modifier toxins', *Toxicon*, 49(2), pp. 124–141. doi: 10.1016/j.toxicon.2006.09.022.

Chang, S. Y. *et al.* (2007) 'Distribution of Kv3.3 potassium channel subunits in distinct neuronal populations of mouse brain', *J Comp Neurol*. 2007/04/21, 502(6), pp. 953–972. doi: 10.1002/cne.21353.

Choe, S. (2002) 'Potassium channel structures', *Nat Rev Neurosci*. 2002/02/12, 3(2), pp. 115–121. doi: 10.1038/nrn727.

Chuck, T. L. *et al.* (2006) 'Comparison between multiple behavioral effects of peripheral ethanol administration in rats: sedation, ataxia, and bradykinesia', *Life Sci*. 2006/02/21, 79(2), pp. 154–161. doi: 10.1016/j.lfs.2005.12.045.

Clay, J. R. (2000) 'Determining K+ channel activation curves from K+ channel currents', *European Biophysics Journal*, 29(7), pp. 555–557. Available at: c:%5CArticles%5Cprinted articles%5CClay_EBpJ29(7)2000.pdf.

Coetzee, W. A. *et al.* (1999) 'Molecular diversity of K+ channels', *Ann N Y Acad Sci*. 1999/07/22, 868, pp. 233–285. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10414301.

Côté, M. P., Murray, L. M. and Knikou, M. (2018) 'Spinal control of locomotion: Individual neurons, their circuits and functions', *Frontiers in Physiology*. doi: 10.3389/fphys.2018.00784.

Cotella, D. *et al.* (2013) 'An evolutionarily conserved mode of modulation of Shaw-like K(+) channels', *FASEB J*. 2012/12/13, 27(4), pp. 1381–1393. doi: 10.1096/fj.12-222778.

Dallas, M. L. *et al.* (2005) 'Localization and function of the Kv3.1b subunit in the rat medulla oblongata: focus on the nucleus tractus solitarii', *J Physiol*. 2004/11/06, 562(Pt 3), pp. 655–672. doi: 10.1113/jphysiol.2004.073338.

Dallas, M. L. *et al.* (2008) 'Voltage-gated potassium currents within the dorsal vagal nucleus: inhibition by BDS toxin', *Brain Res*. 2007/12/01, 1189, pp. 51–57. doi: 10.1016/j.brainres.2007.10.090.

Desai, R. *et al.* (2008) 'Protein kinase C modulates inactivation of Kv3.3 channels', *J Biol Chem*. 2008/06/10, 283(32), pp. 22283–22294. doi: 10.1074/jbc.M801663200.

Deuchars, S. A. *et al.* (2001) 'Properties of interneurones in the intermediolateral cell column of the rat spinal cord: role of the potassium channel subunit Kv3.1',

Neuroscience. 2001/09/22, 106(2), pp. 433–446. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11566512.

Deuchars, S. A. and Lall, V. K. (2015) 'Sympathetic preganglionic neurons: properties and inputs', *Compr Physiol*. 2015/04/17, 5(2), pp. 829–869. doi: 10.1002/cphy.c140020.

Devaux, J. *et al.* (2003) 'Kv3.1b is a novel component of CNS nodes', *J Neurosci*. 2003/06/14, 23(11), pp. 4509–4518. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12805291.

Dodson, P. D., Barker, M. C. and Forsythe, I. D. (2002) 'Two heteromeric Kv1 potassium channels differentially regulate action potential firing.', *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 22(16), pp. 6953–6961. doi: 20026709.

Doyle, D. A. *et al.* (1998) 'The structure of the potassium channel: molecular basis of K+ conduction and selectivity', *Science*. 1998/04/29, 280(5360), pp. 69–77. Available at: http://www.ncbi.nlm.nih.gov/pubmed/9525859.

Du, J. *et al.* (1996) 'Developmental expression and functional characterization of the potassium-channel subunit Kv3.1b in parvalbumin-containing interneurons of the rat hippocampus', *J Neurosci*. 1996/01/15, 16(2), pp. 506–518. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8551335.

Duflocq, A. *et al.* (2011) 'Characterization of the axon initial segment (AIS) of motor neurons and identification of a para-AIS and a juxtapara-AIS, organized by protein 4.1B', *BMC Biology*, 9. doi: 10.1186/1741-7007-9-66.

Erisir, A. *et al.* (1999) 'Function of specific K(+) channels in sustained high-frequency firing of fast-spiking neocortical interneurons', *J Neurophysiol*. 1999/11/24, 82(5), pp. 2476–2489. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10561420.

Espinosa-Medina, I. *et al.* (2018) 'The "sacral parasympathetic": ontogeny and anatomy of a myth', *Clinical Autonomic Research*, pp. 13–21. doi: 10.1007/s10286-017-0478-7.

Espinosa, F. *et al.* (2001) 'Alcohol hypersensitivity, increased locomotion, and spontaneous myoclonus in mice lacking the potassium channels Kv3.1 and Kv3.3', *J Neurosci*. 2001/08/23, 21(17), pp. 6657–6665. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11517255.

Espinosa, F. *et al.* (2004) ‘Increased motor drive and sleep loss in mice lacking Kv3-type potassium channels’, *Genes Brain Behav*. 2004/03/10, 3(2), pp. 90–100. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15005717.

Fernandez, F. R. *et al.* (2003) ‘Inactivation of Kv3.3 Potassium Channels in Heterologous Expression Systems’, *Journal of Biological Chemistry*, 278(42), pp. 40890–40898. doi: 10.1074/jbc.M304235200.

Fineberg, J. D., Ritter, D. M. and Covarrubias, M. (2012) ‘Modeling-independent elucidation of inactivation pathways in recombinant and native A-type Kv channels’, *The Journal of General Physiology*, 140(5), pp. 513–527. doi: 10.1085/jgp.201210869.

Fowler, C. J., Griffiths, D. and de Groat, W. C. (2008) ‘The neural control of micturition’, *Nat Rev Neurosci*. 2008/05/21, 9(6), pp. 453–466. doi: 10.1038/nrn2401.

Goldberg, E. M. *et al.* (2005) ‘Specific functions of synaptically localized potassium channels in synaptic transmission at the neocortical GABAergic fast-spiking cell synapse’, *J Neurosci*. 2005/05/27, 25(21), pp. 5230–5235. doi: 10.1523/JNEUROSCI.0722-05.2005.

Grissmer, S. *et al.* (1992) ‘The Shaw-related potassium channel gene, Kv3.1, on human chromosome 11, encodes the type l K+ channel in T cells’, *J Biol Chem*. 1992/10/15, 267(29), pp. 20971–20979. Available at: http://www.ncbi.nlm.nih.gov/pubmed/1400413.

Grizel, A. V, Glukhov, G. S. and Sokolova, O. S. (2014) ‘Mechanisms of activation of voltage-gated potassium channels’, *Acta Naturae*. 2015/01/06, 6(4), pp. 10–26. Available at: http://www.ncbi.nlm.nih.gov/pubmed/25558391.

de Groat, W. C. (1976) ‘Mechanisms underlying recurrent inhibition in the sacral parasympathetic outflow to the urinary bladder.’, *The Journal of Physiology*, 257(2), pp. 503–513. doi: 10.1113/jphysiol.1976.sp011381.

de Groat, W. C. *et al.* (1982) ‘Parasympathetic preganglionic neurons in the sacral spinal cord’, *Journal of the Autonomic Nervous System*, 5(1), pp. 23–43. doi: 10.1016/0165-1838(82)90087-X.

Gu, Y. *et al.* (2012) ‘Alternative splicing regulates kv3.1 polarized targeting to adjust maximal spiking frequency’, *J Biol Chem*. 2011/11/23, 287(3), pp. 1755–1769. doi: 10.1074/jbc.M111.299305.

Gu, Y., Barry, J. and Gu, C. (2013) ‘Kv3 channel assembly, trafficking and activity are regulated by zinc through different binding sites’, *J Physiol*. 2013/02/20, 591(Pt 10), pp. 2491–2507. doi: 10.1113/jphysiol.2013.251983.

Hall, M. K. *et al.* (2014) ‘N-Linked glycan site occupancy impacts the distribution of a potassium channel in the cell body and outgrowths of neuronal-derived cells’, *Biochim Biophys Acta*. 2013/10/29, 1840(1), pp. 595–604. doi: 10.1016/j.bbagen.2013.10.025.

Hartmann, S. *et al.* (2018) ‘β-Secretase BACE1 promotes surface expression and function of Kv3.4 at hippocampal mossy fiber synapses.’, *The Journal of Neuroscience*, pp. 2643–17. doi: 10.1523/JNEUROSCI.2643-17.2018.

Harvey, M. *et al.* (2012) ‘Impaired long-range synchronization of gamma oscillations in the neocortex of a mouse lacking Kv3.2 potassium channels’, *J Neurophysiol*. 2012/04/28, 108(3), pp. 827–833. doi: 10.1152/jn.00102.2012.

von Hehn, C. A., Bhattacharjee, A. and Kaczmarek, L. K. (2004) ‘Loss of Kv3.1 tonotopicity and alterations in cAMP response element-binding protein signaling in central auditory neurons of hearing impaired mice’, *J Neurosci*. 2004/02/27, 24(8), pp. 1936–1940. doi: 10.1523/JNEUROSCI.4554-03.2004.

Hernández-Pineda, R. *et al.* (1999) ‘Kv3.1-Kv3.2 channels underlie a high-voltage-activating component of the delayed rectifier K+ current in projecting neurons from the globus pallidus.’, *Journal of neurophysiology*, 82(3), pp. 1512–28. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10482766.

Hille, B. (2001) *Ionic channels of excitable membranes, Sinauer Associates Inc Sunderland MA USA*. doi: 10.1002/jnr.490130415.

Hodgkin, A. L. and Huxley, A. F. (1952) ‘Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo’, *J Physiol*. 1952/04/01, 116(4), pp. 449–472. Available at: http://www.ncbi.nlm.nih.gov/pubmed/14946713.

Hoppa, M. B. *et al.* (2014) ‘Control and plasticity of the presynaptic action potential waveform at small CNS nerve terminals’, *Neuron*, 84(4), pp. 778–789. doi: 10.1016/j.neuron.2014.09.038.

Horn, J. P. (2018) ‘The sacral autonomic outflow is parasympathetic: Langley got it right’, *Clinical Autonomic Research*, pp. 181–185. doi: 10.1007/s10286-018-0510-6.

Hurlock, E. C. *et al.* (2009) ‘Rescue of motor coordination by Purkinje cell-targeted restoration of Kv3.3 channels in Kcnc3-null mice requires Kcnc1’, *J Neurosci*. 2009/12/18, 29(50), pp. 15735–15744. doi: 10.1523/JNEUROSCI.4048-09.2009.

Ishikawa, T. *et al.* (2003) ‘Distinct roles of Kv1 and Kv3 potassium channels at the calyx of Held presynaptic terminal’, *J Neurosci*. 2003/11/14, 23(32), pp. 10445–10453. Available at: http://www.ncbi.nlm.nih.gov/pubmed/14614103.

Itri, J. N. *et al.* (2005) ‘Fast delayed rectifier potassium current is required for circadian neural activity’, *Nat Neurosci*. 2005/04/27, 8(5), pp. 650–656. doi: 10.1038/nn1448.

J.A., P. *et al.* (2008) ‘Modelling prefrontal cortex deficits in schizophrenia: Implications for treatment’, *British Journal of Pharmacology*, pp. S465–S470. Available at: http://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=reference&D=emed8&NEWS=N&AN=2008115574.

Jan, L. Y. and Jan, Y. N. (1997) ‘Cloned potassium channels from eukaryotes and prokaryotes’, *Annu Rev Neurosci*. 1997/01/01, 20, pp. 91–123. doi: 10.1146/annurev.neuro.20.1.91.

Jiang, B. *et al.* (2002) ‘Endogenous Kv channels in human embryonic kidney (HEK-293) cells.’, *Molecular and cellular biochemistry*, 238(1–2), pp. 69–79. doi: 10.1023/A:1019907104763.

Jiang, Y. *et al.* (2003) ‘X-ray structure of a voltage-dependent K+ channel’, *Nature*. 2003/05/02, 423(6935), pp. 33–41. doi: 10.1038/nature01580.

Johnston, J., Forsythe, I. D. and Kopp-Scheinpflug, C. (2010) ‘Going native: voltage-gated potassium channels controlling neuronal excitability’, *J Physiol*. 2010/06/04, 588(Pt 17), pp. 3187–3200. doi: 10.1113/jphysiol.2010.191973.

Kaczmarek, L. K. and Zhang, Y. (2017) ‘Kv3 Channels: Enablers of Rapid Firing, Neurotransmitter Release, and Neuronal Endurance’, *Physiological Reviews*, 97(4), pp. 1431–1468. doi: 10.1152/physrev.00002.2017.

Kanda, V. A. *et al.* (2011a) ‘KCNE1 and KCNE2 inhibit forward trafficking of homomeric N-type voltage-gated potassium channels’, *Biophys J*. 2011/09/29, 101(6), pp. 1354–1363. doi: 10.1016/j.bpj.2011.08.015.

Kanda, V. A. *et al.* (2011b) 'KCNE1 and KCNE2 provide a checkpoint governing voltage-gated potassium channel alpha-subunit composition', *Biophys J*. 2011/09/29, 101(6), pp. 1364–1375. doi: 10.1016/j.bpj.2011.08.014.

Keller, J. A. *et al.* (2018) 'Voluntary urination control by brainstem neurons that relax the urethral sphincter', *Nature Neuroscience*. doi: 10.1038/s41593-018-0204-3.

Kopljar, I. *et al.* (2013) 'The ladder-shaped polyether toxin gambierol anchors the gating machinery of Kv3.1 channels in the resting state', *The Journal of General Physiology*, 141(3), pp. 359–369. doi: 10.1085/jgp.201210890.

Kuznetsov, K. I. *et al.* (2012) 'Kv3 channels modulate calcium signals induced by fast firing patterns in the rat retinal ganglion cells', *Cell Calcium*. 2012/07/27, 52(5), pp. 405–411. doi: 10.1016/j.ceca.2012.06.007.

Labro, A. J. *et al.* (2015) 'Kv3.1 uses a timely resurgent K(+) current to secure action potential repolarization', *Nat Commun*. 2015/12/18, 6, p. 10173. doi: 10.1038/ncomms10173.

Lewis, A., McCrossan, Z. A. and Abbott, G. W. (2004) 'MinK, MiRP1, and MiRP2 diversify Kv3.1 and Kv3.2 potassium channel gating', *J Biol Chem*. 2003/12/18, 279(9), pp. 7884–7892. doi: 10.1074/jbc.M310501200.

Li, W., Kaczmarek, L. K. and Perney, T. M. (2001) 'Localization of two high-threshold potassium channel subunits in the rat central auditory system', *J Comp Neurol*. 2001/08/09, 437(2), pp. 196–218. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11494252.

Lien, C. C. and Jonas, P. (2003) 'Kv3 potassium conductance is necessary and kinetically optimized for high-frequency action potential generation in hippocampal interneurons', *J Neurosci*. 2003/03/27, 23(6), pp. 2058–2068. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12657664.

Liu, S. J. and Kaczmarek, L. K. (1998) 'The expression of two splice variants of the Kv3.1 potassium channel gene is regulated by different signaling pathways', *J Neurosci*. 1998/04/29, 18(8), pp. 2881–2890. Available at: http://www.ncbi.nlm.nih.gov/pubmed/9526005.

Llewellyn-Smith, I. J. (2009) 'Anatomy of synaptic circuits controlling the activity of sympathetic preganglionic neurons', *Journal of Chemical Neuroanatomy*, pp. 231–239. doi: 10.1016/j.jchemneu.2009.06.001.

Long, S. B., Campbell, E. B. and Mackinnon, R. (2005) ‘Crystal structure of a mammalian voltage-dependent Shaker family K+ channel’, *Science*. 2005/07/09, 309(5736), pp. 897–903. doi: 10.1126/science.1116269.

Luneau, C. J. *et al.* (1991) ‘Alternative Splicing Contributes to K+ Channel Diversity in the Mammalian Central-Nervous-System’, *Proceedings of the National Academy of Sciences of the United States of America*, 88(9), pp. 3932–3936. doi: DOI 10.1073/pnas.88.9.3932.

Macica, C. M. *et al.* (2003) ‘Modulation of the kv3.1b potassium channel isoform adjusts the fidelity of the firing pattern of auditory neurons’, *J Neurosci*. 2003/02/25, 23(4), pp. 1133–1141. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12598601.

Macica, C. M. and Kaczmarek, L. K. (2001) ‘Casein kinase 2 determines the voltage dependence of the Kv3.1 channel in auditory neurons and transfected cells’, *J Neurosci*. 2001/02/13, 21(4), pp. 1160–1168. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11160386.

Mackel, R. (1979) ‘Segmental and descending control of the external urethral and anal sphincters in the cat.’, *The Journal of Physiology*, 294(1), pp. 105–122. doi: 10.1113/jphysiol.1979.sp012918.

Mannen, T. *et al.* (1977) ‘Preservation of a certain motoneurone group of the sacral cord in amyotrophic lateral sclerosis: its clinical significance’, *Journal of Neurology Neurosurgery and Psychiatry*, 40(5), pp. 464–469. doi: 10.1136/jnnp.40.5.464.

Markram, H. *et al.* (2015) ‘Reconstruction and Simulation of Neocortical Microcircuitry’, *Cell*, 163(2), pp. 456–492. doi: 10.1016/j.cell.2015.09.029.

Martina, M., Yao, G. L. and Bean, B. P. (2003) ‘Properties and functional role of voltage-dependent potassium channels in dendrites of rat cerebellar Purkinje neurons’, *J Neurosci*. 2003/07/05, 23(13), pp. 5698–5707. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12843273.

Matsukawa, H. *et al.* (2003) ‘Motor dysfunction and altered synaptic transmission at the parallel fiber-Purkinje cell synapse in mice lacking potassium channels Kv3.1 and Kv3.3’, *J Neurosci*. 2003/08/22, 23(20), pp. 7677–7684. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12930807.

McCrossan, Z. A. *et al.* (2003) ‘MinK-related peptide 2 modulates Kv2.1 and Kv3.1 potassium channels in mammalian brain’, *J Neurosci*. 2003/09/05, 23(22), pp. 8077–8091. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12954870.

McDonald, A. J. and Mascagni, F. (2006) ‘Differential expression of Kv3.1b and Kv3.2 potassium channel subunits in interneurons of the basolateral amygdala’, *Neuroscience*. 2006/01/18, 138(2), pp. 537–547. doi: 10.1016/j.neuroscience.2005.11.047.

Mende, M. *et al.* (2016) ‘Sensory-Derived Glutamate Regulates Presynaptic Inhibitory Terminals in Mouse Spinal Cord’, *Neuron*, 90(6), pp. 1189–1202. doi: 10.1016/j.neuron.2016.05.008.

Milsom, I., Stewart, W. and Thuroff, J. (2000) ‘The prevalence of overactive bladder’, in *American Journal of Managed Care*.

Monaco, V., Ghionzoli, A. and Micera, S. (2010) ‘Age-related modifications of muscle synergies and spinal cord activity during locomotion’, *J Neurophysiol*. 2010/08/06, 104(4), pp. 2092–2102. doi: 10.1152/jn.00525.2009.

Moreno, H. *et al.* (1995) ‘Thalamocortical projections have a K+ channel that is phosphorylated and modulated by cAMP-dependent protein kinase’, *J Neurosci*. 1995/08/01, 15(8), pp. 5486–5501. Available at: http://www.ncbi.nlm.nih.gov/pubmed/7643197.

Morgan, C. W. *et al.* (1991) ‘Axon collaterals indicate broad intraspinal role for sacral preganglionic neurons.’, *Proceedings of the National Academy of Sciences of the United States of America*, 88(15), pp. 6888–92. doi: 10.1073/pnas.88.15.6888.

Muona, M. *et al.* (2015) ‘A recurrent de novo mutation in KCNC1 causes progressive myoclonus epilepsy’, *Nature Genetics*, 47(1), pp. 39–46. doi: 10.1038/ng.3144.

Noga, B. R. *et al.* (1987) ‘The role of Renshaw cells in locomotion: antagonism of their excitation from motor axon collaterals with intravenous mecamylamine’, *Exp Brain Res*. 1987/01/01, 66(1), pp. 99–105. Available at: http://www.ncbi.nlm.nih.gov/pubmed/3582539.

Nowak, A. *et al.* (2011) ‘Kv3.1b and Kv3.3 channel subunit expression in murine spinal dorsal horn GABAergic interneurones’, *J Chem Neuroanat*. 2011/03/29, 42(1), pp. 30–38. doi: 10.1016/j.jchemneu.2011.02.003.

Oliver, D. *et al.* (2004) ‘Functional conversion between A-type and delayed rectifier K+ channels by membrane lipids’, *Science*. 2004/03/20, 304(5668), pp. 265–270. doi: 10.1126/science.1094113.

Oliver, K. L. *et al.* (2017) 'Myoclonus epilepsy and ataxia due to KCNC1 mutation: Analysis of 20 cases and K^+channel properties', *Annals of Neurology*, 81(5). doi: 10.1002/ana.24929.

Olsen, T. *et al.* (2018) 'Kv3 K^+currents contribute to spike-timing in dorsal cochlear nucleus principal cells', *Neuropharmacology*, 133. doi: 10.1016/j.neuropharm.2018.02.004.

Ozaita, A. *et al.* (2002) 'Differential subcellular localization of the two alternatively spliced isoforms of the Kv3.1 potassium channel subunit in brain', *J Neurophysiol*. 2002/07/02, 88(1), pp. 394–408. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12091563.

Parameshwaran, S., Carr, C. E. and Perney, T. M. (2001) 'Expression of the Kv3.1 potassium channel in the avian auditory brainstem.', *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 21(2), pp. 485–494.

Parekh, P. K. *et al.* (2018) 'Antimanic efficacy of a novel Kv3 potassium channel modulator', *Neuropsychopharmacology*, 43(2), pp. 435–444. doi: 10.1038/npp.2017.155.

Parsons, B. A. and Drake, M. J. (2011) 'Animal models in overactive bladder research', *Handbook of Experimental Pharmacology*, 202, pp. 15–43. doi: 10.1007/978-3-642-16499-6_2.

Perez-Escudero, A. *et al.* (2014) 'idTracker: tracking individuals in a group by automatic identification of unmarked animals', *Nat Methods*. 2014/06/02, 11(7), pp. 743–748. doi: 10.1038/nmeth.2994.

Perney, T. M. *et al.* (1992) 'Expression of the mRNAs for the Kv3.1 potassium channel gene in the adult and developing rat brain', *J Neurophysiol*. 1992/09/01, 68(3), pp. 756–766. Available at: http://www.ncbi.nlm.nih.gov/pubmed/1432046.

Persson, K. *et al.* (1998) 'Spinal and peripheral mechanisms contributing to hyperactive voiding in spontaneously hypertensive rats.', *The American journal of physiology*, 275(4 Pt 2), pp. R1366-73. Available at: http://www.ncbi.nlm.nih.gov/pubmed/9756570.

Ponce, A. *et al.* (1997) 'K+ channel subunit isoforms with divergent carboxy-terminal sequences carry distinct membrane targeting signals', *J Membr Biol*. 1997/10/06, 159(2), pp. 149–159. Available at: http://www.ncbi.nlm.nih.gov/pubmed/9307441.

Pongs, O. (1999) ‘Voltage-gated potassium channels: from hyperexcitability to excitement’, *FEBS Lett*. 1999/06/22, 452(1–2), pp. 31–35. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10376673.

Rettig, J. *et al.* (1992) ‘Characterization of a Shaw-related potassium channel family in rat brain’, *EMBO J*. 1992/07/01, 11(7), pp. 2473–2486. Available at: http://www.ncbi.nlm.nih.gov/pubmed/1378392.

Ritter, D. M. *et al.* (2012) ‘Modulation of Kv3.4 channel N-type inactivation by protein kinase C shapes the action potential in dorsal root ganglion neurons’, *J Physiol*. 2011/11/09, 590(Pt 1), pp. 145–161. doi: 10.1113/jphysiol.2011.218560.

Rosato-Siri, M. D. *et al.* (2015) ‘A Novel Modulator of Kv3 Potassium Channels Regulates the Firing of Parvalbumin-Positive Cortical Interneurons’, *J Pharmacol Exp Ther*. 2015/06/19, 354(3), pp. 251–260. doi: 10.1124/jpet.115.225748.

Rosenberg, A. B. *et al.* (2018) ‘Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding’, *Science*, 360(6385), pp. 176–182. doi: 10.1126/science.aam8999.

Rowan, M. J. M. and Christie, J. M. (2017) ‘Rapid State-Dependent Alteration in Kv3 Channel Availability Drives Flexible Synaptic Signaling Dependent on Somatic Subthreshold Depolarization’, *Cell Reports*, 18(8), pp. 2018–2029. doi: 10.1016/j.celrep.2017.01.068.

Rowan, M. J., Tranquil, E. and Christie, J. M. (2014) ‘Distinct Kv channel subtypes contribute to differences in spike signaling properties in the axon initial segment and presynaptic boutons of cerebellar interneurons’, *J Neurosci*. 2014/05/09, 34(19), pp. 6611–6623. doi: 10.1523/JNEUROSCI.4208-13.2014.

Rudy, B. *et al.* (1999) ‘Contributions of Kv3 channels to neuronal excitability’, *Ann N Y Acad Sci*. 1999/07/22, 868, pp. 304–343. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10414303.

Rudy, B. and McBain, C. J. (2001) ‘Kv3 channels: voltage-gated K+ channels designed for high-frequency repetitive firing’, *Trends Neurosci*. 2001/08/17, 24(9), pp. 517–526. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11506885.

Sack, J. T., Shamotienko, O. and Dolly, J. O. (2008) ‘How to Validate a Heteromeric Ion Channel Drug Target: Assessing Proper Expression of Concatenated Subunits: Figure 1.’, *The Journal of General Physiology*, 131(5), pp. 415–420. doi: 10.1085/jgp.200709939.

Santer, R. M. *et al.* (2002) ‘Differential susceptibility to ageing of rat preganglionic neurones projecting to the major pelvic ganglion and of their afferent inputs’, *Auton Neurosci*. 2002/04/03, 96(1), pp. 73–81. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11926170.

Sasaki, S. I., Uchino, H. and Uchino, Y. (1994) ‘Axon branching of medullary expiratory neurons in the lumbar and the sacral spinal cord of the cat’, *Brain Research*, 648(2), pp. 229–238. doi: 10.1016/0006-8993(94)91122-3.

Shefchyk, S. J. (2001) ‘Sacral spinal interneurones and the control of urinary bladder and urethral striated sphincter muscle function’, *J Physiol*. 2001/05/15, 533(Pt 1), pp. 57–63. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11351013.

Siroky, M. B. (2004) ‘The aging bladder’, *Rev.Urol.*, 6 Suppl 1(1523–6161 (Print)), pp. S3–S7.

Song, P. *et al.* (2005) ‘Acoustic environment determines phosphorylation state of the Kv3.1 potassium channel in auditory neurons’, *Nat Neurosci*. 2005/09/02, 8(10), pp. 1335–1342. doi: 10.1038/nn1533.

Song, Z. M. *et al.* (2006) ‘Developmental changes in the expression of calbindin and potassium-channel subunits Kv3.1b and Kv3.2 in mouse Renshaw cells’, *Neuroscience*. 2006/02/08, 139(2), pp. 531–538. doi: 10.1016/j.neuroscience.2005.12.048.

Stephan, A. H. *et al.* (2013) ‘A Dramatic Increase of C1q Protein in the CNS during Normal Aging’, *Journal of Neuroscience*, 33(33), pp. 13460–13474. doi: 10.1523/JNEUROSCI.1333-13.2013.

Suskind, A. M. (2017) ‘The Aging Overactive Bladder: a Review of Aging-Related Changes from the Brain to the Bladder’, *Current Bladder Dysfunction Reports*, pp. 42–47. doi: 10.1007/s11884-017-0406-7.

Taskin, B. *et al.* (2015) ‘Biophysical characterization of KV3.1 potassium channel activating compounds’, *European Journal of Pharmacology*, 758, pp. 164–170. doi: 10.1016/j.ejphar.2015.03.061.

Thorneloe, K. S. and Nelson, M. T. (2003) ‘Properties and molecular basis of the mouse urinary bladder voltage-gated K+ current’, *J Physiol*. 2003/04/08, 549(Pt 1), pp. 65–74. doi: 10.1113/jphysiol.2003.039859.

Todd, A. J. (2010) ‘Neuronal circuitry for pain processing in the dorsal horn’, *Nature Reviews Neuroscience*, 11(12), pp. 823–836. doi: 10.1038/nrn2947.

Todd, A. J. (2017) ‘Identifying functional populations among the interneurons in laminae I-III of the spinal dorsal horn’, *Molecular Pain*. doi: 10.1177/1744806917693003.

Wang, L. *et al.* (2010) ‘GABAB Mediated Regulation of Sympathetic Preganglionic Neurons: Pre- and Postsynaptic Sites of Action’, *Frontiers in Neurology*, 1. doi: 10.3389/fneur.2010.00142.

Wang, L. Y. *et al.* (1998) ‘Contribution of the Kv3.1 potassium channel to high frequency firing in mouse auditory neurones’, *Journal of Physiology*, 509(1), pp. 183–194. doi: 10.1111/j.1469-7793.1998.183bo.x.

Wang, Z., Robertson, B. and Fedida, D. (2007) ‘Gating currents from a Kv3 subfamily potassium channel: Charge movement and modification by BDS-II toxin’, *Journal of Physiology*, 584(3), pp. 755–767. doi: 10.1113/jphysiol.2007.140145.

Weiser, M. *et al.* (1994) ‘Differential expression of Shaw-related K+ channels in the rat central nervous system’, *J Neurosci*. 1994/03/01, 14(3 Pt 1), pp. 949–972. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8120636.

Weiser, M. *et al.* (1995) ‘The potassium channel subunit KV3.1b is localized to somatic and axonal membranes of specific populations of CNS neurons’, *J Neurosci*. 1995/06/01, 15(6), pp. 4298–4314. Available at: http://www.ncbi.nlm.nih.gov/pubmed/7790912.

Xu, J. *et al.* (1995) ‘Assembly of voltage-gated potassium channels. Conserved hydrophilic motifs determine subfamily-specific interactions between the alpha-subunits’, *J Biol Chem*. 1995/10/20, 270(42), pp. 24761–24768. Available at: http://www.ncbi.nlm.nih.gov/pubmed/7559593.

Xu, M. *et al.* (2007) ‘The axon-dendrite targeting of Kv3 (Shaw) channels is determined by a targeting motif that associates with the T1 domain and ankyrin G’, *J Neurosci*. 2007/12/21, 27(51), pp. 14158–14170. doi: 10.1523/JNEUROSCI.3675-07.2007.

Xu, M. *et al.* (2010) ‘Kinesin I transports tetramerized Kv3 channels through the axon initial segment via direct binding’, *J Neurosci*. 2010/11/26, 30(47), pp. 15987–16001. doi: 10.1523/JNEUROSCI.3565-10.2010.

Yanagi, M. *et al.* (2014) ‘Kv3.1-containing K(+) channels are reduced in untreated schizophrenia and normalized with antipsychotic drugs’, *Mol Psychiatry*. 2013/05/01, 19(5), pp. 573–579. doi: 10.1038/mp.2013.49.

Yoshimura, N. *et al.* (1990) ‘α1-adrenergic receptor-mediated excitation from the locus coeruleus of the sacral parasympathetic preganglionic neuron’, *Life Sciences*, 47(9), pp. 789–797. doi: 10.1016/0024-3205(90)90551-2.

Zettel, M. L. *et al.* (2007) ‘Age-related decline in Kv3.1b expression in the mouse auditory brainstem correlates with functional deficits in the medial olivocochlear efferent system’, *J Assoc Res Otolaryngol*. 2007/04/25, 8(2), pp. 280–293. doi: 10.1007/s10162-007-0075-x.

Zhang, Y. and Kaczmarek, L. K. (2016) ‘Kv3.3 potassium channels and spinocerebellar ataxia’, *Journal of Physiology*, pp. 4677–4684. doi: 10.1113/JP271343.

Zhang, Y.et al. (2016) ‘Kv3.3 Channels Bind Hax-1 an Arp2/3 to Assemble a Stable Local Actin Network that Regulates Channel Gating’, *Cell*, 165(2), pp. 434-448. doi:10.1016/j.cell.2016.02.009.

Appendices

Appendix I - HEK.py

```
import numpy as np
import pandas as pd
import stf
from matplotlib import pyplot as plt
import Tkinter, tkFileDialog
import os
from scipy.optimize import curve_fit
import math
import scipy.signal as sp


class HEK():
#Create core df for each cell
#Ask user for folder where control access resistance files are kept
#Ask user for folder where control currents are
#Ask user for folder where drug access resistances are
#Ask user for folder where drug controls are
#Open access resistance file in folder and analyse
#Open current files in folder and analyse
#Make figure of recordings for user to evaluate which to exclude
#From df make figure for cell
#Save df to csv and when enough cells acquired, assimilate the averages into a new df
#plot graphs and do stats on this

# To Do

# Save figures
# Save dfs
# Create function that compares control vs drug
# Give plots titles, data labels and axes labels
# Need to add drug concentration
# Check appropriate global variables are reset each time a function is called
# Add access resistances to control and drug dataframes-done
    plt.rcParams.update({'font.size': 6})
    base_directory= "C:\\"
    file_path=0
    control_Ras=0
    control_currents=0
    control_times=0
    drug_times=0
    inact_times=0
    drug_Ras=0
    drug_currents=0
    inact_currents=0
    inact_ras=0
    drug_inact_ras=0
    drug_inact_currents=0
    times=0
    name=0
    intracellular=0
    cell_type= 0
    rs= 0
    compensation=0
    capacitance=0
    date=0
    cell_number=0

    pyplot_markers=["o", "v", "^", "<", ">", "s", "p", "8"]
    pyplot_colours=['b', 'g', 'r', 'c', 'm', 'y', 'k', 'w']


    control_Ras_files=0
    control_currents_files = 0
    drug_Ras_files = 0
    drug_currents_files =0
    inact_currents_files=0
    inact_ras_files=0
    drug_inact_currents_files=0
    drug_inact_ras_files=0

    ra_av_traces=[0,0,0]
    ra_max_currents=[0,0,0]
    ra_access_resistances=[0,0,0]
    percent_diffs=[0,0,0]
    holding_currents=[0,0,0]
    holding_current=0
    current_traces=0
    plateau_currents=0
    peak_currents=0
    peak_current_times=0

    amp_unit=0
    peak_leak=0
    plat_leak=0
    samp_int=0
    start=0
    end=0
    index=["-90", "-80", "-70", "-60", "-50", "-40", "-30", "-20", "-10", "0", "10", "20", "30", "40", "50"]
    inact_index=["-70", "-50", "-30", "-10", "10", "30"]
    inact_pre_step=0
```

```
inact_df=pd.DataFrame(index=inact_index)
control_conductance_df=pd.DataFrame(index=index)
activation_df=pd.DataFrame(index=index)
peak_df=0
GHKplat_df=0
GHKpeak_df=0
access_resistance_df=0
holding_current_df=0
taus=0
tau_df=0
root=0
HC_df=0
writer=0
fig1=0
fig2=0
leak_df=pd.DataFrame(index=["-90"])
subplots=0
exclusions=0
# TO DO- show access resistance earlier as part of figure-done
# save file with date-done
# add exclusion-done-needs thorough testing
# find some way to define between peak and plateau in final excel document-done
# add more ticks to graphs
# be able to handle any amound of files-done needs testing
def index_refresh(self):
    self.index = ["-90", "-80", "-70", "-60", "-50", "-40", "-30", "-20", "-10", "0", "10", "20", "30", "40", "50"]
    self.inact_index=["-70", "-50", "-30", "-10", "10", "30"]

def basic_workflow(self):
    self.df_refresh()
    self.leak_df=pd.DataFrame(index=["-90"])
    self.con_df_refresh()
    self.index_refresh()
    self.user_info()
    self.extra_info()
    self.times=self.control_times
    self.file_lists()
    self.writer = pd.ExcelWriter(self.base_directory + "//"+ self.intracellular+"_"+self.condition+
                        "_"+self.cell_type+"_"+self.date+"_"+self.cell_number+ ".xlsx", engine="xlsxwriter")
    self.plot_creation()
    if self.control_Ras_files != 0:

        self.activation(self.control_currents, self.control_currents_files, "control", self.control_Ras,
                    self.control_Ras_files, self.control_times)
    else:
        print "no activation files"

    if self.drug_Ras_files!=0:
        self.activation(self.drug_currents, self.drug_currents_files, "drug", self.drug_Ras,
                    self.drug_Ras_files, self.drug_times)
    else:
        print "no drug files"
    self.con_df_refresh()

    if self.inact_ras_files!=0:
        self.inactivation(self.inact_currents, self.inact_currents_files, "inact", self.inact_ras,
                      self.inact_ras_files, self.inact_times)
    else:
        print "no inactivation currents"

    if self.drug_inact_ras_files!=0:
        self.inactivation(self.drug_inact_currents, self.drug_inact_currents_files, "druginact", self.drug_inact_ras,
                      self.drug_inact_ras_files, self.drug_inact_times)
    else:
        print "no drug inactivation currents"

    self.writer.save()
    print "saved"

    fig3_figname=self.base_directory+"/"+"CurrentsandG.png"
    fig4_figname = self.base_directory + "/" + "Stability.png"
    self.fig3.savefig(fig3_figname)
    self.fig4.savefig(fig4_figname)
    self.fig3.show()
    self.fig4.show()

def activation(self, currents, files, condition, Ras, ras_files, times):
        self.df_refresh()
        self.times=times
        self.condition = condition
        self.peak_df_func(self.index)
        self.HC_df = pd.DataFrame(index=self.index)
        self.current_analysis(currents, files,
                          self.activation_df, 220, 230, "activation")  # change these
        self.access_resistance(Ras, ras_files,
                           condition, self.activation_df)
        self.activation_df, self.peak_df = self.append_dataframe(self.activation_df, self.peak_df)
        self.GHKplat_df, self.GHKpeak_df = self.append_dataframe(self.GHKplat_df, self.GHKpeak_df)
        self.exclusion(self.activation_df)
        self.mean_columns = raw_input("Which columns would you like to average?").split(",")
        self.control_conductance_col = raw_input("What time is baseline time? Just write number").split(",")

        self.df_analysis(self.activation_df, condition)
        self.df_analysis(self.peak_df, condition)
        self.df_analysis(self.GHKplat_df, condition)
        self.df_analysis(self.GHKpeak_df, condition)

        self.activation_df=self.pd_concat(self.activation_df, self.peak_df)
        """self.activation_df = pd.concat([self.activation_df, self.peak_df], axis=1)
```

```
        self.activation_df = pd.concat([self.activation_df, self.tau_df], axis=1)
        self.activation_df = pd.concat([self.activation_df, self.HC_df], axis=1)"""
        self.GHKplat_df=self.pd_concat(self.GHKplat_df, self.GHKpeak_df)


        self.index.append("Access_resistance")
        self.index.append("Percentage_difference")
        self.index.append("Holding_current")
        self.activation_df = self.activation_df.reindex(self.index)
        self.GHKplat_df = self.GHKplat_df.reindex(self.index)

        self.activation_df=self.boltzmann_fits(self.activation_df)
        self.GHKplat_df=self.boltzmann_fits(self.GHKplat_df)


        self.activation_df.to_excel(self.writer, sheet_name=condition)
        self.GHKplat_df.to_excel(self.writer, sheet_name="GHK"+condition)

        self.index_refresh()
        if condition=="drug":
            self.times = [int(x) + int(self.control_times[-1]) for x in self.times]
            self.times = [str(x) for x in self.times]
        self.plots(self.activation_df, self.times, self.index)

        #add a boltzmann fit function here

    def pd_concat(self, dataframe1, dataframe2):
        dataframe1=pd.concat([dataframe1, dataframe2], axis=1)
        dataframe1=pd.concat([dataframe1, self.tau_df], axis=1)
        dataframe1=pd.concat([dataframe1, self.HC_df], axis=1)
        return dataframe1

    def inactivation(self, currents, files, condition, Ras, ras_files, times):
        self.df_refresh()
        self.times=times
        self.condition = condition
        self.index_refresh()
        self.inact_df=pd.DataFrame(index=self.inact_index)
        self.peak_df_func(self.inact_index)
        inact_start = 0
        inact_end = 0
        if self.inact_pre_step != "\n":
            inact_start = float(self.inact_pre_step) * 1000 + 970
            inact_end = float(self.inact_pre_step) * 1000 + 980
        else:
            inact_start = (20 * 1000) + 970
            inact_end = (20 * 1000) + 980
        self.HC_df = pd.DataFrame(index=self.inact_index)
        self.current_analysis(currents, files,
                              self.inact_df, inact_start, inact_end, "inact")

        self.access_resistance(Ras, ras_files,
                               condition, self.inact_df)
        self.inact_df, self.peak_df = self.append_dataframe(self.inact_df, self.peak_df)
        self.GHKplat_df, self.GHKpeak_df = self.append_dataframe(self.GHKplat_df, self.GHKpeak_df)

        self.exclusion(self.inact_df)

        self.mean_columns = raw_input("Which columns would you like to average?").split(",")
        self.control_conductance_col = raw_input("What time is baseline time? Just write number").split(",")

        self.df_analysis(self.inact_df, condition)
        self.df_analysis(self.peak_df, condition)
        self.df_analysis(self.GHKplat_df, condition)
        self.df_analysis(self.GHKpeak_df, condition)

        self.inact_df = pd.concat([self.inact_df, self.peak_df], axis=1)
        self.inact_df = pd.concat([self.inact_df, self.HC_df], axis=1)

        self.GHKplat_df= pd.concat([self.GHKplat_df, self.GHKpeak_df], axis=1)
        self.GHKplat_df= pd.concat([self.GHKplat_df, self.HC_df], axis=1)
        try:
            self.GHKplat_df = pd.concat([self.GHKplat_df, self.pre_step_df], axis=1)
            self.inact_df = pd.concat([self.inact_df, self.pre_step_df], axis=1)
        except:
            print "Couldn't concat pre_step_df"

        self.inact_index.append("Access_resistance")
        self.inact_index.append("Percentage_difference")
        self.inact_index.append("Holding_current")

        self.inact_df = self.inact_df.reindex(self.inact_index)
        self.GHKplat_df= self.GHKplat_df.reindex(self.inact_index)

        self.inact_df = self.boltzmann_fits(self.inact_df)
        self.GHKplat_df = self.boltzmann_fits(self.GHKplat_df)


        # self.inact_df = pd.concat([self.inact_df, self.tau_df], axis=1)  # these taus dont seem right-fixed
        # self.inact_df.reindex(self.index) dont need # this could throw up issues with exclusions-fixed exclsions changes index, refreshed before next condition
        self.inact_df.to_excel(self.writer, sheet_name=condition)
        self.GHKplat_df.to_excel(self.writer, sheet_name="GHK"+condition)

        self.index_refresh()
        self.plots(self.inact_df, self.times, self.inact_index)
        self.df_refresh()

    def plt_close(self):
        for n in range(0, 4):
```

```
        plt.close(n+1)

    def peak_df_func(self, index):
        self.peak_df=pd.DataFrame(index=index)
        self.GHKpeak_df=pd.DataFrame(index=index)
        self.GHKplat_df=pd.DataFrame(index=index)
        self.pre_step_df=pd.DataFrame(index=index)

    def df_refresh(self):

        self.activation_df = pd.DataFrame(index=self.index)
        self.inact_df = pd.DataFrame(index=self.inact_index)

    def con_df_refresh(self):
        self.control_conductance_df=pd.DataFrame(index=self.index)

    def folder_select(self):
        self.root=Tkinter.Tk()
        self.root.withdraw()
        self.root.directory= self.base_directory
        self.file_path=tkFileDialog.askdirectory(initialdir=self.base_directory)


    def user_info(self):

        #need to add reseting parameters here
        print "Select the cell folder"
        self.folder_select()
        self.base_directory=self.file_path

        print "Which folder are control access resistance files in?"
        self.folder_select()
        self.control_Ras=self.file_path
        print "What folder are control currents in"
        self.folder_select()
        self.control_currents=self.file_path
        #self.control_times=raw_input("What times were recordings obtained at in mins? Enter in format #,#,#").split(",")
        print "Which folder are drug access resistance files in?"
        self.folder_select()
        self.drug_Ras = self.file_path
        print "Which folder are drug currents in"
        self.folder_select()
        self.drug_currents = self.file_path
        print "Which folder are inactivation access resistance files in"
        self.folder_select()
        self.inact_ras=self.file_path
        print "Which folder are inactivation currents in"
        self.folder_select()
        self.inact_currents=self.file_path
        print "Which folder are drug inactivation access resistance files in"
        self.folder_select()
        self.drug_inact_ras = self.file_path
        print "Which folder are drug inactivation currents in"
        self.folder_select()
        self.drug_inact_currents = self.file_path
        mistake=raw_input("Type y if you want to repeat, enter to continue")
        if mistake=="y":
            self.user_info()
        else:
            pass
    def extra_info(self):
        self.control_times=raw_input("What times were recordings obtained at in mins? Enter in format #,#,#").split(",")
        self.drug_times=raw_input("What times were drug recordings obtained at in mins? Enter in format #,#,#").split(",")
        self.inact_times = raw_input("What times were inact recordings obtained at in mins? Enter in format #,#,#").split(",")
        self.drug_inact_times = raw_input("What times were drug_inact recordings obtained at in mins? Enter in format #,#,#").split(",")
        self.inact_pre_step = raw_input("Length of inact pre-step in seconds. Use 20.0255 for 3.1 prot. if no V out")
        self.condition= raw_input("What is the condition? e.g control, AUT2006")
        self.intracellular= raw_input("What type of intracellular used?")
        self.cell_type= raw_input("What cell type?")
        self.current_channel=raw_input("Analyse leak subtraction only, y/n?")
        #self.rs= raw_input("What Rs values were recorded? Enter in format #,#,#").split(",")
        self.compensation=raw_input("What % compensation was used? If one value will be used for all").split(",")
        #self.capacitance=raw_input("What was the capacitance? If only one value, will be used for all").split(",")
        self.date=raw_input("Date of recording")
        self.cell_number=raw_input("Cell number")
        mistake2 = raw_input("Type y if you want to repeat, enter to continue")
        if mistake2 == "y":
            self.extra_info()
        else:
            pass

    def file_lists(self):
        self.control_Ras_files=0
        self.control_currents_files=0
        self.drug_Ras_files=0
        self.drug_currents_files=0
        self.inact_currents_files=0
        self.inact_ras_files=0
        self.drug_inact_currents_files=0
        self.drug_inact_ras_files=0
        try:
            self.control_Ras_files=os.listdir(self.control_Ras)
        except:
            print "no folder provided"
        try:
            self.control_currents_files = os.listdir(self.control_currents)
        except:
            print "no folders provided"
        try:
```

```
            self.drug_Ras_files = os.listdir(self.drug_Ras)
        except:
            print "no folder provided"
        try:
            self.drug_currents_files = os.listdir(self.drug_currents)
        except:
            print "no folder provided"
        try:
            self.inact_ras_files=os.listdir(self.inact_ras)
        except:
            print "no folder provided"
        try:
            self.inact_currents_files=os.listdir(self.inact_currents)
        except:
            print "no folder provided"
        try:
            self.drug_inact_ras_files=os.listdir(self.drug_inact_ras)
        except:
            print "no folder provided"
        try:
            self.drug_inact_currents_files=os.listdir(self.drug_inact_currents)
        except:
            print "no folder provided"
    def check_file(self, operator):
        size=4000
        size_warning=0
        trace_size=stf.get_size_trace()
        if operator == "==":
            if trace_size==size:
                pass

            else:
                size_warning = raw_input(
                    "Incorrect file size detected, check correct files in folder. Type y to continue"
                    "or n to redo file selection after updating")

        elif operator==">":
            if trace_size>size:
                pass
            else:
                size_warning = raw_input(
                    "Incorrect file size detected, check correct files in folder. Type y to continue"
                    "or n to redo file selection after updating")

        if size_warning=="n":
                self.file_lists()
        elif size_warning=="y":
                pass


    # Access resistance calculator
    def access_resistance(self, folder, files, condition, dataframe): # do this for each folder, maybe add condition for drug/control
        #access files in folder
        #look through files open each one in turn and analyse
        self.ra_av_traces = []
        self.ra_max_currents = []
        self.ra_access_resistances = []
        self.percent_diffs = []
        self.access_resistance_df = 0
        for file in files:

            file=folder+"/"+file
            file=file.replace("/", '\\')
            file=file.encode("ascii", "ignore")
            stf.file_open(file)
            self.check_file("==")
            traces = []

            for n in range(0, int(stf.get_size_channel())):
                trace = stf.get_trace(n)
                traces.append(trace)
            traces=np.array(traces)
            average_trace=np.mean(traces, axis=0)
            self.ra_av_traces.append(average_trace)
            baseline=average_trace[0:int(2/stf.get_sampling_interval())]
            baseline_av = np.mean(baseline)
            max_current = np.max(average_trace) - baseline_av
            self.ra_max_currents.append(max_current)
            access_resistance=(10/max_current)*1000
            self.ra_access_resistances.append(access_resistance)
            stf.close_this()

        index = ["Access_resistance", "Percentage_difference"]
        #columns=["0", "1", "2"]#this here might be limiting factor
        columns=[]
        columns2=[]
        df_columns=dataframe.columns
        for n in range(0, len(df_columns)):
            columns.append(df_columns[n])
            columns2.append(self.peak_df.columns[n])

        for n in self.ra_access_resistances:
            self.percent_diffs.append(((self.ra_access_resistances[0]-n)/self.ra_access_resistances[0])*100)

        self.access_resistance_df=pd.DataFrame([self.ra_access_resistances, self.percent_diffs], index=index,
                                               columns=columns)
        self.access_resistance_df2=pd.DataFrame([self.ra_access_resistances, self.percent_diffs], index=index,
                                               columns=columns2)
```

```
    for n in range(0, len(df_columns)):

        half_max=(np.max(self.peak_df.iloc[:, n])/2)
        self.subplots[n].text(250, half_max*1.8, "Ra= %s Mohm" %(str(round(self.ra_access_resistances[n],1))))
        self.subplots[n].text(250, half_max*1.4, "Diff= %s " % (str(round(self.percent_diffs[n], 1))))
        self.subplots[n].text(250, half_max*0.8, "H.curr.= %s" % (str(round(self.average_hc[n], 1))))

    fig_name=self.base_directory+"/"+condition+"currents.png"
    self.fig1.savefig(fig_name)
    self.fig1.show()


def differentiate(self, trace):
    self.array=np.arange(len(trace)-1)
    for n in range(0, len(trace) - 2):
        self.array[n] = (trace[n + 1] - trace[n]) / stf.get_sampling_interval()

def get_protocol(self):
    trace_number=stf.get_size_channel(1)
    last_trace=0
    pre_step_start=0
    pre_step_end=0
    last_trace=stf.get_trace(trace_number-1, 1)

    self.differentiate(last_trace)
    if trace_number<10:
        pre_step_start=np.argmax(self.array[0:int(1500/stf.get_sampling_interval())])
        pre_step_end=pre_step_start+2000
        last_trace=stf.get_trace(0,1)
        self.differentiate(last_trace)
    threshold=0
    if stf.get_sampling_interval()<0.1:
        threshold=2000
    elif stf.get_sampling_interval()==0.1:
        threshold=800
    self.peaks=[]
    self.troughs=[]

    window=500/stf.get_sampling_interval()
    numOfwindows=int((len(self.array)/window))
    for n in range(0, numOfwindows-1):
        win_start=int(n*window)
        win_end=int((n+1)*window)
        peak=np.max(self.array[win_start:win_end])
        peak_time=np.argmax(self.array[win_start:win_end])+win_start
        trough=np.min(self.array[win_start:win_end])
        trough_time=np.argmin(self.array[win_start:win_end])+win_start
        if peak>threshold:
            self.peaks.append([peak, peak_time])
        else:
            pass
        if trough<-threshold:
            self.troughs.append([trough, trough_time])
        else:
            pass
    if np.shape(self.peaks)==np.shape(self.troughs):
        numOfstim=np.shape(self.peaks)[0]

    else:
        print "unequal peaks and troughs-inactivation file"
    start=int(self.peaks[-1][-1])
    end=int(self.troughs[-1][-1])

    self.peak_start=int(start+23.5)
    self.peak_end=int(self.peak_start+((end-start)/2))
    self.plateau_end=int(end-62)
    self.plateau_start=int(self.plateau_end-(10/stf.get_sampling_interval()))
    self.deactivation_start=end+10

    self.raw_outputs=[]
    self.pre_step_plats = []
    self.pre_step_peaks = []
    self.pre_step_peak_times =[]
    for n in range(0, trace_number):
        raw_output=0
        pre_step_plat=0
        pre_step_peak=0
        if trace_number>10:
            raw_output=np.mean(stf.get_trace(n, 1)[start:end])
        elif trace_number<10:
            raw_output = np.mean(stf.get_trace(n, 1)[pre_step_start:pre_step_end])
            pre_step_plat = np.mean(stf.get_trace(n, 0)[pre_step_end-110:pre_step_end-10])
            pre_step_peak_time = np.argmax(stf.get_trace(n, 0)[pre_step_start+24:pre_step_end-1000])
            pre_step_peak_time=(pre_step_start+24)+pre_step_peak_time
            pre_step_peak=np.mean(stf.get_trace(n, 0)[pre_step_peak_time:pre_step_peak_time+10])
            self.pre_step_plats.append(pre_step_plat)
            self.pre_step_peaks.append(pre_step_peak)
            self.pre_step_peak_times.append(pre_step_peak_time)

        self.raw_outputs.append(raw_output)

    diff1=self.raw_outputs[2]-self.raw_outputs[1]
    diff2=self.raw_outputs[1]-self.raw_outputs[0]
    diffs=[diff1, diff2]
    step_size=np.mean(diffs)
    holding=self.raw_outputs[0]
    if np.allclose(10, step_size, atol=3) is True:
        self.steps=np.arange(-90, -90+10*trace_number, 10)
    elif np.allclose(20, step_size, atol=3) is True:
```

```
        self.steps=np.arange(-70, -70+20*trace_number, 20)
    #self.index=[str(n) for n in self.steps]          # this should work intheory but quite weak
    #self.inact_index=self.index
def current_analysis(self, folder, files, dataframe, start, end, condition):

    #amend this to have a start and end for current to be analysed-done
    #for activation is 180-190, for inactivation is 590-600-maybe include peak for inact-done
    #create new df for inactivation, add inactivation file requests-done
    #access resi to be calculated for inact so add to that function-done
    #self.samp_int=stf.get_sampling_interval()
    self.fits=[]
    self.average_hc=[]
    count=0
    self.fig1=plt.figure()
    self.tau_df=0

    self.tau_df=pd.DataFrame(index=self.index[-9:])

    self.subplots=[]
    for file in files:
        channel_no=0
        self.current_traces = []
        self.leak_current_traces =[]
        self.holding_current = 0
        self.plateau_currents = []
        self.peak_currents =[]
        self.leak_plateau_currents =[]
        self.leak_peak_currents = []
        self.peak_current_times=[]
        self.leak_peak_current_times = []
        self.holding_currents = []
        self.deact_starts=[]
        self.leak_deact_starts=[]
        file = folder + "/" + file
        file = file.replace("/", '\\')
        file = file.encode("ascii", "ignore")
        stf.file_open(file)
        self.check_file(">")

        if stf.get_size_recording()==1:
            self.peak_start = int((end - 198.14) / stf.get_sampling_interval())
            print self.peak_start
            self.peak_end = int((end - 188.14) / stf.get_sampling_interval())
            self.plateau_start=int(start/stf.get_sampling_interval())
            self.plateau_end=int(end/stf.get_sampling_interval())

        else:
            self.get_protocol()

        if self.current_channel=="y":
            channel_no=2
        else:
            channel_no=0
        for n in range(0, int(stf.get_size_channel())):
            trace=stf.get_trace(n, channel_no)
            self.holding_current=np.mean(trace[0:int(10/stf.get_sampling_interval())])
            self.holding_currents.append(self.holding_current)
            self.leak_current_traces.append(trace) #this is raw trace fo GHK
            trace = trace - self.holding_current
            self.current_traces.append(trace) #this is hc subtracted trace for OHM

        self.av_hc=np.mean(self.holding_currents)
        self.average_hc.append(self.av_hc)
        # currents at 180-190 ms
        #for n in self.current_traces[:8]: #this need modifying
        ###OHM
        current_index=0
        for n in self.current_traces:

            plateau_current=np.mean(n[self.plateau_start:self.plateau_end])
            self.plateau_currents.append(plateau_current)
            max_time=np.argmax(n[self.peak_start:self.peak_end])
            peak_current = np.mean(n[self.peak_start+(max_time):self.peak_start+(max_time+10)])
            self.peak_currents.append(peak_current)#might need to change tghis to -198.65 due to capacitance spike in

            peak_current_time=np.argmax(n[self.peak_start:self.peak_end])
            peak_current_time=(peak_current_time+self.peak_start)*stf.get_sampling_interval()
            self.peak_current_times.append(peak_current_time)
            deact_start=np.argmax(n[11580:11655])+11580
            self.deact_starts.append(deact_start)
            current_index=current_index+1
        ###GHK
        current_index = 0
        for n in self.leak_current_traces:
            plateau_current = np.mean(n[self.plateau_start:self.plateau_end])
            self.leak_plateau_currents.append(plateau_current)
            max_time = np.argmax(n[self.peak_start:self.peak_end])
            peak_current = np.mean(n[self.peak_start + (max_time):self.peak_start + (max_time + 10)])
            self.leak_peak_currents.append(peak_current)  # might need to change tghis to -198.65 due to capacitance spike in

            peak_current_time = np.argmax(n[self.peak_start:self.peak_end])
            peak_current_time = (peak_current_time + self.peak_start) * stf.get_sampling_interval()
            self.leak_peak_current_times.append(peak_current_time)
            deact_start = np.argmax(n[11580:11655])+11580
            self.leak_deact_starts.append(deact_start)
            current_index = current_index + 1


        if np.max(self.peak_currents)>100:
```

```
    self.amp_unit="pA"
else:
    self.amp_unit="nA"
##ADD a check here to see if leak_df is empty- if empty then currents appended as norm

if "inact" in condition:
        if sum(self.leak_df.loc["-90"]) != 0:
            if "drug" in self.condition:
                self.peak_leak=self.leak_df["drugPeak Leak"]
                self.plat_leak=self.leak_df["drugPlat Leak"]

            else:
                self.peak_leak = self.leak_df["controlPeak Leak"]
                self.plat_leak = self.leak_df["controlPlat Leak"]
            if self.peak_leak[0]<-3:
                self.peak_leak=self.peak_leak[0]/1e03
                self.plat_leak=self.plat_leak[0]/1e03
            else:
                self.peak_leak = self.peak_leak[0]
                self.plat_leak = self.plat_leak[0]
        else:
            print "Leak not subtracted for GHK analysis"

else:
        self.peak_leak=self.leak_peak_currents[0]
        self.plat_leak=self.leak_plateau_currents[0]
        self.leak_df[self.condition+"Peak Leak"]=self.peak_leak
        self.leak_df[self.condition+"Plat Leak"]=self.plat_leak
        print "Not inact-leak subtracted"
x=[-90, 0]
y=[self.peak_leak, 0]
y3=[self.plat_leak, 0]
peak_fit=np.polyfit(x, y, 1)
plat_fit=np.polyfit(x, y3, 1)

x2=[int(z) for z in self.index]
peak_y=[self.leaksub(x, *peak_fit) for x in x2]
plat_y=[self.leaksub(x, *plat_fit) for x in x2]
print peak_y
print plat_y
for n in self.leak_peak_currents:
        index = int(self.leak_peak_currents.index(n))
        if "inact" in condition:
            self.leak_peak_currents[index] = n-peak_y[-2]
            self.leak_plateau_currents[index] = self.leak_plateau_currents[index]-plat_y[-2]
            try:
                self.pre_step_peaks[index]=self.pre_step_peaks[index]-peak_y[index]
                self.pre_step_plats[index]=self.pre_step_plats[index]-plat_y[index]
            except:
                print "no V out trace present"
        else:
            self.leak_peak_currents[index]= n - peak_y[index]
            self.leak_plateau_currents[index] = self.leak_plateau_currents[index] - plat_y[index]

            #test

self.peak_df["Peak "+self.times[count]+" mins"] = self.peak_currents
#column_header=0
dataframe["Plateau "+self.times[count]+" mins"]=self.plateau_currents
self.GHKpeak_df["Peak "+self.times[count]+" mins"] = self.leak_peak_currents
self.GHKplat_df["Plateau "+self.times[count]+" mins"]=self.leak_plateau_currents
try:
    self.pre_step_df["Prestep Peak"+self.times[count]+" mins"]=self.pre_step_peaks
    self.pre_step_df["Prestep Plateau" + self.times[count] + " mins"] = self.pre_step_plats
except:
    print "Probs not inact file"
    #column_header=str(self.drug_times[count])+" mins"
    #self.drug_df[str(count)]=self.plateau_currents  # change count back to column header


# maybe condense this into one figure with 3 subplots
folder_size=np.ceil(np.sqrt(len(files)))

ax1=self.fig1.add_subplot(folder_size, folder_size, count+1)
self.subplots.append(ax1)
self.subplots[count].set_ylabel("Current")
self.subplots[count].set_ylim(bottom=self.av_hc, top=np.max(self.peak_currents)*1.8)
self.subplots[count].set_xlabel("Time, ms")
x=list(range(stf.get_size_trace()))
x=np.array(x)
x = x[0:int(self.plateau_end+ (380 / stf.get_sampling_interval()))]
x = x * stf.get_sampling_interval()
index=0
for n in self.current_traces:

    y= n[0:int((self.plateau_end+(380/stf.get_sampling_interval())))]
    self.subplots[count].plot(x, y, "k")
    a=(self.peak_current_times[index])

    b=self.peak_currents[index]
    try:
        c=np.array(self.pre_step_peak_times)*stf.get_sampling_interval()
        d=self.pre_step_peaks
        self.subplots[count].scatter(c, d)
    except:
        print "Couldn't plot activation steps for inactivation trace"
    self.subplots[count].scatter(a, b)
    index=index+1
```

```
        self.subplots[count].plot([self.peak_start*stf.get_sampling_interval(), self.peak_start*stf.get_sampling_interval()],
                                  [0, self.peak_currents[-1]], "r") # to show peak window
        self.subplots[count].plot([self.peak_end*stf.get_sampling_interval(), self.peak_end*stf.get_sampling_interval()],
                                  [ 0 , self.peak_currents[-1]], "r") # to show peak window
        self.subplots[count].plot([self.plateau_start*stf.get_sampling_interval(), self.plateau_start*stf.get_sampling_interval()],
                                  [0, self.peak_currents[-1]], "b") #to show average window
        self.subplots[count].plot([self.plateau_end*stf.get_sampling_interval(), self.plateau_end*stf.get_sampling_interval()],
                                  [0, self.peak_currents[-1]], "b") # to show average window

        # need to add a conditional here to do fit just for Kv3.4
        # if cell type contains Kv3.4
        if "inact" not in condition:

            self.act_exp_fit(count+2)

            if "3.4" in self.cell_type.lower():

                self.exponential_fit(count+2)

        stf.close_this()

        self.HC_df["HC"+str(count)] = self.holding_currents
        count=count+1

    columns=[] # a fix
    columns2=[]
    for n in dataframe.columns:
        columns.append(n)
    for n in self.peak_df.columns:
        columns2.append(n)
    self.holding_current_df = pd.DataFrame([self.average_hc], index=["Holding_current"], #
                                           columns=columns)
    self.holding_current_df2= pd.DataFrame([self.average_hc], index=["Holding_current"], #
                                           columns=columns2)#limiting factor



def exclusion(self, dataframe): #check this works and doesnt just modify a copy of the df

        self.exclusions=raw_input("Do you want to exclude any voltage steps? Enter values").split(",")
        #maybe just hold onto to exlusion list and use to edit final dataframe
        if self.exclusions[0]!="\n":
            for n in self.exclusions:
                dataframe.drop(n)
                self.peak_df.drop(n)
            self.index=dataframe.index
        #return dataframe
            #why did i want Nan surely better to drop completely

def conductance(self, dataframe, condition, classifier):

        Gmax = 0
        GHKmax = 0
        Correctgmax = 0
        true_vm=0

        access_resistances = np.array(dataframe.loc["Access_resistance"])
        ra_mean = np.mean(access_resistances[self.mean_start:self.mean_end])
        access_resistances[-2]=ra_mean

        for n in dataframe.columns[:-1]:

            index = np.argwhere(dataframe.columns == n)

            Runcomp = (access_resistances[index]*1e06) * ((100-float(self.compensation[0])) / 100)


            column_title=str(n)[-7:]
            conductances = []
            true_vms=[]
            VEs=[]
            correctgs=[]
            GHK_perm=[]
            gk=0
            conductance=0
            correctg=0
            F=96485.0
            R=8.314
            k=1.38e-23
            T=293.15  #20degC
            q=1.6e-19
            Zs=1.0
            Ki=151.25
            Ko=4.0
            Ek=-94.0/1e03
            test_step=float(40.0/1e03)

            for row in dataframe.itertuples():
                if row[0] == "Access_resistance" or row[0] == "Percentage_difference" or row[0] == "Holding_current":
                    conductances.append("Nan")
                    true_vms.append("Nan")
                    VEs.append("Nan")
                    correctgs.append("Nan")
                    GHK_perm.append("Nan")

                else:
                    I=dataframe.loc[row[0], n]
                    if self.amp_unit == "pA":
```

```
            I=I/1e12
        else:
            I=I/1e09
        VE=I*Runcomp

        if"inact" in condition:
            true_vm = ((test_step) - (VE))
            conductance = I / ((test_step) - (Ek))
            correctg = I / (float(true_vm) - (Ek))
            first = np.exp((float(q * (true_vm - float(Ek)))) / (float(k * T))) - float(1)
            second = np.exp((float(q * true_vm)) / (k * T)) - 1
            #first=np.exp((true_vm-float(Ek))/25.0)-1.0
            #second=np.exp(true_vm/25.0)-1.0
            gk = I / ((true_vm/25.0) * (first / second))

            """first = (Zs ** 2) * (((true_vms[0]) * (F ** 2)) / (R * T))
            second = (Ki - Ko) * np.exp(float(-Zs * F * true_vms[0]) / float(R * T))
            third = 1 - np.exp(float(-Zs * F * true_vms[0]) / float(R * T))
            P = I / (first * (second / third))"""
        else:
            true_vm=((float(row[0]) / 1e03) - (VE))
            conductance = I / ((float(row[0])/1e03) - (Ek))
            correctg= I / (float(true_vm) - (Ek))
            first = np.exp((float(q * (true_vm - float(Ek)))) / (float(k * T))) - float(1)
            second=np.exp((float(q*true_vm))/(k*T))-float(1)

            gk=I/(true_vm*(first/second))
            """first = (Zs ** 2) * (((true_vm) * (F ** 2)) / (R * T))
            second = (Ki - Ko) * np.exp(float(-Zs * F * true_vm) / float(R * T))
            third = 1 - np.exp(float(-Zs * F * true_vm) / float(R * T))
            gk = I / (first * (second / third))"""
        VEs.append(VE)
        true_vms.append(true_vm)
        correctgs.append(correctg)
        conductances.append(conductance)
        GHK_perm.append(gk)

dataframe[classifier + " G(S)" + "of" + column_title] = conductances
dataframe[classifier + "TrueVm" + "of" + column_title] = true_vms
dataframe[classifier + "VErr" + "of" + column_title] = VEs
dataframe[classifier + "CorrectG(S)" + "of" + column_title] = correctgs
dataframe[classifier + "GHK_perm" + "of" + column_title] = GHK_perm

conductance_len=len(conductances)


conductances2 = [x for x in conductances if type(x) != str]
GHK_con = [x for x in GHK_perm if type(x) !=str]
Correctg= [x for x in correctgs if type(x) !=str]
if "drug" in condition:

    last_control_column=0
    GHK_control=0
    Correctg_control=0
    Gs = []

    for c in self.control_conductance_df.columns:
        if classifier=="Plat.":

            if "Plat." in c:
                Gs.append(c)
        elif classifier=="Pk.":
            if "Pk." in c:
                Gs.append(c)

    control_conductance_col=self.control_conductance_col
    conditional="G(S)of "+control_conductance_col[0]
    GHK_conditional="GHK_permof "+control_conductance_col[0]
    Correctg_conditional= "CorrectGof "+control_conductance_col[0]
    for g in Gs:
        if conditional in g:
            last_control_column=self.control_conductance_df[g]
        elif GHK_conditional in g:
            GHK_control=self.control_conductance_df[g]
        elif Correctg_conditional in g:
            Correctg_control = self.control_conductance_df[g]
    #last_control_column=[x for x in last_control_column if type(x)!= str]
    if "inact" in condition:
        Gmax = last_control_column[0]
        GHKmax = GHK_control[0]
        Correctgmax = Correctg_control[0]
    else:
        Gmax=np.max(last_control_column[5:])
        GHKmax=np.max(GHK_control[5:])
        Correctgmax=np.max(Correctg_control[5:])

else:
    if "inact" in condition:
        self.control_conductance_df = self.control_conductance_df.reindex(self.inact_index)
        Gmax = conductances2[0]
        GHKmax = GHK_con[0]
        Correctgmax = Correctg[0]
    else:
        self.control_conductance_df = self.control_conductance_df.reindex(self.index)
        Gmax = np.max(conductances2[5:])
        GHKmax=np.max(GHK_con[5:])
        Correctgmax = np.max(Correctg[5:])

    self.control_conductance_df[classifier + " G(S)" + "of" + column_title] = conductances2
```

```
            self.control_conductance_df[classifier + " GHK_perm" + "of" + column_title] = GHK_con
            self.control_conductance_df[classifier + " CorrectG" + "of" + column_title] = Correctg

        conductances=np.array(conductances2)
        gGmax=conductances/Gmax
        GHK_norm=GHK_con/GHKmax
        CorrectgGmax=Correctg/Correctgmax
        len_diff=conductance_len-len(gGmax)
        nan_sub=len_diff*("Nan").split(",")
        gGmax=np.append(gGmax, nan_sub)
        GHK_norm=np.append(GHK_norm, nan_sub)
        CorrectgGmax=np.append(CorrectgGmax, nan_sub)

        dataframe[classifier + " G/Gmax" + "of" + column_title] = gGmax
        dataframe[classifier + " GHK_norm" + "of" + column_title] = GHK_norm
        dataframe[classifier + " CorrectgGmax" + "of" + column_title] = CorrectgGmax


def df_analysis(self, dataframe, condition):#change condition to df as uses less code
    #itertuples lopp for row in itertuples-calculate mean, sem, std

    means=[]
    stds=[]
    self.mean_start=0
    self.mean_end=0
    mean_columns=self.mean_columns
    if mean_columns=="\n":
        mean_columns=["0"]

    mean_columns=[int(x)+1 for x in mean_columns]
    mean_start=np.min(mean_columns)
    mean_end=np.max(mean_columns)+1
    for row in dataframe.itertuples():
        if row[0]== "Access_resistance" or row[0]== "Percentage_difference" or row[0]=="Holding_current":
            means.append("Nan")
            stds.append("Nan")
        else:
            mean = np.mean(row[mean_start:mean_end])
            std = np.std(row[mean_start:mean_end])
            means.append(mean)
            stds.append(std)

    classifier=0
    for n in dataframe.columns:
        if "Peak" in n:
            classifier="Pk."
        else:
            classifier="Plat."

    mean_columns=tuple(mean_columns)
    place_markers=classifier+" Mean of columns"
    for n in mean_columns:
        place_markers=place_markers+" %d "

    dataframe[place_markers %(mean_columns)]=means
    dataframe[classifier+" Std"]=stds

    self.conductance(dataframe, condition, classifier)


def exponential_fit(self, fig_num): # need to define a changin number for figure number becuase it just keeps building upon itself
    x=0
    y=0
    index=6
    plt_num=1
    plt.figure()
    self.taus=[]

    for n in self.current_traces[-9:]:
        start=int((self.peak_current_times[index]+1)/stf.get_sampling_interval())
        end=int(start+(100/stf.get_sampling_interval()))
        x=list(range(end-start))
        x=np.array(x)
        x=x*stf.get_sampling_interval()
        y=n[start:end]
        popt = [0, 0, 0]
        try:
            popt, pcov = curve_fit(self.func, x, y, p0=(self.peak_currents[index], 10, self.plateau_currents[index]), maxfev=2000)
        except:
            popt=[0,0,0]
            print "Couldn't fit inactivation"
        plt.subplot(3,3,plt_num)
        plt.plot(x, y)
        plt.plot(x, self.func(x, *popt))
        index=index+1
        plt_num+=1
        self.taus.append(popt[1])

    self.taus=np.array(self.taus)
    self.taus=1/self.taus
    self.tau_df["inact_tau"+str(fig_num-2)]=self.taus
    figname = self.base_directory + "/inact_fit" + str(fig_num)+".png"
    plt.savefig(figname)
    plt.show()

def act_exp_fit(self, fig_num):
    x = 0
    y = 0
    index = 6
```

```
    plt_num = 1
    plt.figure()
    self.act_taus = []
    start=0
    end=0

    for n in self.current_traces[-9:]:
        start = int(self.peak_start)
        if "3.4" in self.cell_type.lower():
            end=int(self.peak_current_times[index]/stf.get_sampling_interval())
        else:
            end = int(self.peak_end)
        x = list(range(end - start))
        x = np.array(x)
        x = x * stf.get_sampling_interval()
        y = n[start:end]
        popt = [0, 0, 0]
        try:
            popt, pcov = curve_fit(self.act_func, x, y, p0=(self.peak_currents[index], 1, -(self.peak_currents[index]-n[start])), maxfev=2000)
        except:
            popt=[0,0,0]
            print "no fit"
        plt.subplot(3, 3, plt_num)
        plt.plot(x, y)
        plt.plot(x, self.act_func(x, *popt))
        index = index+1
        plt_num += 1
        self.act_taus.append(popt[1])

    self.act_taus = np.array(self.act_taus)
    self.act_taus = 1 / self.act_taus
    self.tau_df["act_tau" + str(fig_num - 2)] = self.act_taus
    figname = self.base_directory + "/act_fit" + str(fig_num) + ".png"
    plt.savefig(figname)
    plt.show()

def act_func(self, x, a, b, c):
    return 1-(a*np.exp(-b*x)+c)

def func(self, x, a, b, c):
    return a*np.exp(-b*x)+c   # a should be initial amplitude taken from peak current, c should be plateau current
                              # tau should be set as 10
def leaksub(self, x, m, c):
    return (m*x)+c


def append_dataframe(self, dataframe1, dataframe2): # for some reason doenst modify global dataframe
    #evaluate shapes of dataframes and adjust accordingly
    #error here but not sure what issue is so quick patch would be to concantenate df with itself along axis 1
        dataframe1=dataframe1.append(self.access_resistance_df)
        dataframe1=dataframe1.append(self.holding_current_df)
        dataframe2 = dataframe2.append(self.access_resistance_df2)
        dataframe2 = dataframe2.append(self.holding_current_df2)
        return dataframe1, dataframe2

def plot_creation(self):

    self.fig3 = plt.figure()
    self.plat_plot = self.fig3.add_subplot(2, 2, 1)
    self.plat_plot.set_ylabel("Peak Current")
    self.plat_plot.set_xlabel("Voltage, mV")
    self.plat_plot.set_xticks(np.arange(-90, 50, 10))

    self.peak_plot=self.fig3.add_subplot(2, 2, 3)
    self.peak_plot.set_ylabel("Plateau Current")
    self.peak_plot.set_xlabel("Voltage, mv")
    self.peak_plot.set_xticks(np.arange(-90, 50, 10))

    self.ggmax_plot = self.fig3.add_subplot(2, 2, 2)
    self.ggmax_plot.set_ylabel("G/Gmax")
    self.ggmax_plot.set_xlabel("Voltage, mV")
    self.ggmax_plot.set_yticks(np.arange(0, 1.2, 0.2))
    self.ggmax_plot.set_xticks(np.arange(-90, 50, 10))
    self.ggmax_plot.set_ylim(bottom=0, top=1.3)

    self.fig4 = plt.figure()

    self.plat_stab = self.fig4.add_subplot(3, 2, 1)
    self.plat_stab.set_ylabel("Plateau Current (50mV)")

    self.peak_stab = self.fig4.add_subplot(3, 2, 2)
    self.peak_stab.set_ylabel("Peak current (50mv)")
    self.peak_stab.set_xlabel("Time")

    self.access_stab = self.fig4.add_subplot(3, 2, 3)
    self.access_stab.set_ylabel("Ra (Mohm)")
    self.access_stab.set_xlabel("Time")

    self.inact_stab = self.fig4.add_subplot(3, 2, 5)
    self.inact_stab.set_ylabel("Inact. Tau (ms))")
    self.inact_stab.set_xlabel("Time")

    self.act_stab = self.fig4.add_subplot(3, 2, 4)
    self.act_stab.set_ylabel("Act. Tau (ms))")
    self.act_stab.set_xlabel("Time")

def plots(self, dataframe, times, index):
    # add a plot of all currents versus voltage to see if cell is good enough
    # maybe make this one figure
```

```
        #dataframe mining
        peak_columns = []
        plateau_columns = []
        inact_tau_columns=[]
        act_tau_columns = []
        peak_ggmax_columns=[]
        plat_ggmax_columns=[]
        for n in dataframe.columns:
            if "Peak " in n:
                peak_columns.append(n)
            if "Plateau " in n:
                plateau_columns.append(n)
            if "act_tau" in n:
                act_tau_columns.append(n)
            if "inact_tau" in n:
                inact_tau_columns.append(n)
            if "Pk. G/Gmax" in n:
                peak_ggmax_columns.append(n)
            if "Plat. G/Gmax" in n:
                plat_ggmax_columns.append(n)

        #peak currents
        x = []
        for row in dataframe.itertuples():
            if len(row[0])<4 and row[0]!= "V2" and row[0] != "k":
                x.append(int(row[0]))
        line_colour=0
        maxV=0
        if "drug" in self.condition:
            line_colour = "r"
        else:
            line_colour="k"

        if "inact" in self.condition:
            maxV = self.inact_index[-1]
        else:
            maxV = self.index[-1]

        for n in range(0, len(peak_columns)):
            data_column=np.argwhere(dataframe.columns==peak_columns[n]).flatten()
            y=dataframe.iloc[0:len(index), data_column] #this is where inact plot would fall down
            self.plat_plot.plot(x, y, line_colour+self.pyplot_markers[n], label="t= "+times[n]+self.condition, linestyle="-") #add condition here and set each control as
one colur but drug as diff ones
        self.plat_plot.legend()

        #plateau currents
        for n in range(0, len(plateau_columns)):
            data_column=np.argwhere(dataframe.columns==plateau_columns[n]).flatten()
            y=dataframe.iloc[0:len(index), data_column]
            self.peak_plot.plot(x, y, line_colour+self.pyplot_markers[n], label="t= "+times[n]+self.condition, linestyle="-")
        self.peak_plot.legend()

        #peak and plat ggmax
        x=[]
        y=0
        for row in dataframe.itertuples():
            if len(row[0])<4 and row[0]!="V2" and row[0]!="k":
                x.append(int(row[0]))
        if "3.4" in self.cell_type.lower():
            for n in range(0, len(peak_ggmax_columns)-1):
                data_column=np.argwhere(dataframe.columns==peak_ggmax_columns[n]).flatten()
                y=dataframe.iloc[0:len(index), data_column]
                y=np.array(y, dtype=float).flatten()
                self.ggmax_plot.scatter(x, y, label="t= "+times[n]+self.condition)
                self.ggmax_plot.legend()
        else:
            for n in range(0, len(plat_ggmax_columns)-1):
                data_column = np.argwhere(dataframe.columns == plat_ggmax_columns[n]).flatten()
                y = dataframe.iloc[0:len(index), data_column]
                y = np.array(y, dtype=float).flatten()
            self.ggmax_plot.scatter(x, y, label="t= "+times[n]+self.condition)
            self.ggmax_plot.legend()



        #STABILITY PLOTS
        #Plateau current
        x=times
        y=dataframe.loc[maxV, plateau_columns]
        self.plat_stab.scatter(x, y)
        #Peak current
        x=times
        y = dataframe.loc[maxV, peak_columns]
        self.peak_stab.scatter(x, y)
        #Access resistance
        x=times
        y=dataframe.loc["Access_resistance", peak_columns]
        self.access_stab.scatter(x, y)
        #Inactivation
        if len(inact_tau_columns)!= 0:
            x = times
            y = dataframe.loc[maxV, inact_tau_columns]
            try:
                self.inact_stab.scatter(x, y)
            except:
                print "Couldn't plot inact tau"
        #Activation
        if len(act_tau_columns)!=0:
```

```
        x = times
        y = dataframe.loc[maxV, act_tau_columns]
        try:
            self.act_stab.scatter(x, y)
        except:
            print "Couldn't plot act_tau"

    def boltzmann_fits(self, dataframe):

        #scrape each df for normalised GHK, correct conductance-maybe ask which time
        #do all columns
        G = []
        CorrectG = []
        norm_GHK = []
        self.boltz_df=0
        self.boltz_df=pd.DataFrame(index=["bottom","Gmax", "V2", "k"])

        for n in dataframe.columns:
            if "G/Gmax" in n:
                G.append(n)
            elif "CorrectgGmax" in n:
                CorrectG.append(n)
            elif "GHK_norm" in n:
                norm_GHK.append(n)
        x=dataframe.index
        x=[int(n) for n in x if len(n)<4]
        if "inact" not in self.condition:
            x=x[2:-1]
        else:
            x=x#this could have implications for inactivation?
        # create loop through G, correctG, GHK norm, and do curve fit, append curve_fit variables df,
        # concat this df to main df
        cond=[G, CorrectG, norm_GHK]

        plt.figure()
        plt_num=1
        rowcols=np.ceil(np.sqrt((len(self.times)+1)*6))
        for con in cond:
            for n in con:

                y=dataframe[n]
                if "inact" not in self.condition:
                    y=y[2:len(x)+2]
                else:
                    y=y[:len(x)]
                y=[float(z) for z in y]
                popt, pcov= curve_fit(self.boltz_func, x, y, p0=(0, 1, 0, 5), maxfev=1500)
                self.boltz_df[n]=popt
                plt.subplot(rowcols, rowcols, plt_num)
                plt.scatter(x, y)
                plt.plot(x, self.boltz_func(x, *popt), "r")
                plt_num+=1
        plt.show()

        dataframe=pd.concat([dataframe, self.boltz_df], axis=0)
        return dataframe

    def boltz_func(self, x, bottom, gmax, V2, k):
        G=bottom+(gmax-bottom)/(1+np.exp((V2-x)/k))
        return G

    def sing_AP(self):
        onsets=[58]
        peaks=[70]
        self.bigAP_df = pd.DataFrame(
            index=["Max_current", "Peak_Latency", "Onset_Latency", "bottom", "top", "v2", "k", "A", "tau", "c"])
        self.trace_figure = plt.figure()
        for n in range(0, stf.get_size_channel()):
            trace=stf.get_trace(n, 0)
            max = np.ceil(np.sqrt(stf.get_size_channel()))
            self.trace_plot = self.trace_figure.add_subplot(max, max, n+1)
            self.single_AP(onsets, peaks, trace, n)

        path = stf.get_filename()[:-4]
        self.bigAP_df.to_csv(path +"single"+ ".csv")
        show=raw_input("WOuld you like to see deact and act plots? type y for yes n for n")
        if show=="y":
            self.act_figure.show()
            self.deact_figure.show()
        else:
            pass
        self.trace_figure.show()
    def trains(self):
        """ pass list of onsets and peaks """
        #peaks
        first=[3096, 3929, 4933, 5907, 7314, 8716, 9941,
            11131, 12131]
        second=[2379, 2717, 2995, 3377, 3623, 3964, 4333,
            4616, 4960, 5270, 5575, 5943, 6286, 6582,
            6961, 7321, 7599, 7962, 8281, 8583, 8957,
            9265, 9590, 9969, 10342, 10700, 11121, 11455,
            11791, 12167]
        third=[ 2303, 2501, 2713, 2919, 3130, 3332, 3535,
            3770, 3994, 4273, 4514, 4727, 4966, 5235,
            5494, 5724, 5949, 6176, 6457, 6703, 6945,
            7171, 7450, 7691, 7940, 8152, 8375, 8650,
            8901, 9133, 9357, 9632, 9894, 10128, 10357,
            10584, 10865, 11112, 11347, 11571, 11839, 12098]
```

```
    #onsets
    first_on=[3087,  3914,  4920,  5890,  7300,  8700,  9925,
    11115,  12116]

    second_on=[2368,  2706,  2978,  3325,  3608,  3946,  4317,
     4605,  4943,  5255,  5559,  5927,  6268,  6568,
     6946,  7301,  7583,  7949,  8266,  8563,  8935,
     9251,  9572,  9950, 10327, 10685, 11103, 11440,
     11774,  12151]

    third_on=[2286,  2488,  2695,  2905,  3114,  3317,  3518,
     3755,  3979,  4255,  4495,  4713,  4946,  5218,
     5475,  5709,  5931,  6158,  6442,  6685,  6925,
     7153,  7435,  7675,  7920,  8130,  8354,  8625,
     8884,  9115,  9342,  9613,  9877, 10108, 10334,
     10562, 10844, 11092, 11326, 11550, 11817, 12077]

    onsets=[first_on, second_on, third_on]
    peaks=[first, second, third]
    self.trace_figure = plt.figure()

    for n in range(0, stf.get_size_channel()):
        trace=stf.get_trace(n, 0)
        max=np.ceil(np.sqrt(stf.get_size_channel()))
        self.trace_plot = self.trace_figure.add_subplot(max, max, n+1)
        self.single_AP(onsets[n], peaks[n], trace, n)
    show = raw_input("WOuld you like to see deact and act plots? type y for yes n for n")
    if show=="y":
        self.act_figure.show()
        self.deact_figure.show()
    else:
        pass
    self.trace_figure.show()
    #define each peak AP point
    #measure latency from each point to current peak(get value of peak)
    # use single APfunction


def single_AP(self, onsets, peaks, trace, index):
    """onset for single should = [58], peak=[70], type channel as string e.g "3.4" """
    path = stf.get_filename()[:-4]

    self.AP_df = pd.DataFrame(index=["Max_current","Peak_Latency", "Onset_Latency", "bottom", "top", "v2", "k", "A", "tau", "c"])

    self.act_figure=plt.figure()
    self.deact_figure=plt.figure()

    y=stf.get_trace(index, 1)
    y=y-y[0]
    I=trace

    mean=0
    x=[]
    #get voltage out file and normalise to current trace
    #make voltage start from 0 then find factor between peak
    """if len(onsets) == 1:
        for n in range(0, stf.get_size_channel()):
            x.append(stf.get_trace(n))

        mean = np.mean(x, axis=0)
    else:"""
    mean = I
    mean = mean - mean[0]
    for n in onsets:
        onset_index=onsets.index(n)
        onset=n
        peak=peaks[onset_index]
        rowcols=np.ceil(np.sqrt(len(onsets)))
        act_plt = self.act_figure.add_subplot(rowcols, rowcols, onset_index+1)
        deact_plt = self.deact_figure.add_subplot(rowcols,rowcols,onset_index+1)
        samp_int = stf.get_sampling_interval()


        window = peak + int((10 / samp_int))
        max_current = np.max(mean[peak:window])
        peak_latency = np.argmax(mean[peak:window])*samp_int
        max_current_t = np.argmax(mean[peak:window]) + peak
        onset_latency=(max_current_t-onset)*samp_int
        self.trace_plot.scatter((max_current_t*samp_int), max_current)
        # rising fit - having issues with fits
        rising = mean[onset:max_current_t]

        #
        # rising = sp.medfilt(rising)
        x = np.arange(len(rising))
        rising = np.delete(rising, [2, 3, 4, 5, 6, 7, 8])
        x = np.delete(x, [2, 3, 4, 5, 6, 7, 8])
        x=x*samp_int
        act_popt=[0,0,0,0]
        act_plt.scatter(x, rising)
        try:
            act_popt, pcov = curve_fit(self.boltz_func, x, rising, p0=(1, 1, 1, 1), maxfev=1000)
            act_plt.plot(x, self.boltz_func(x, *act_popt))
        except:
            print "no AP_act fit"
            act_popt=[0,0,0,0]
```

```
            # decay_fit
            decay = mean[max_current_t:window+int((5/samp_int))]
            x = np.arange(len(decay))*samp_int
            deact_popt=[0,0,0]
            deact_plt.plot(x, decay)
            try:
                deact_popt, pcov = curve_fit(self.func, x, decay, p0=(max_current, 1, 0), maxfev=1000)
                deact_plt.plot(x, self.func(x, *deact_popt))
                deact_popt[1]=1/deact_popt[1]
            except:
                print "no AP_deact fit"
                deact_popt=[0,0,0]

            ap=[max_current, peak_latency, onset_latency]
            for n in act_popt:
                ap.append(n)
            for n in deact_popt:
                ap.append(n)
            if stf.get_size_channel()>3:
                self.AP_df[str(index)]=ap
            else:
                self.AP_df[str(onset_index)]=ap

        maxV = np.max(y)
        maxI = np.max(mean)
        ratio = maxV / maxI
        y = y / ratio
        y = sp.medfilt(y)
        I = sp.medfilt(mean)
        y = y[0:16000]
        I = I[0:16000]
        x = np.arange(len(y))
        x=x*samp_int# up to 16000
        self.trace_plot.plot(x, y)
        self.trace_plot.plot(x, I)



        if stf.get_size_channel() > 3:
            self.bigAP_df=pd.concat([self.bigAP_df, self.AP_df], axis=1)
        else:
            self.AP_df.to_csv(path +str(index)+ ".csv")
        #define single peak point
    #onset at 58
    #peak is at about 70
    #create 5 ms window after peak in other channel


"""def deactivation_fit(self):
    index=0
    for n in self.deact_starts:
        trace=self.current_traces[index]
        fit=trace[int(n):int(n+(1/stf.get_sampling_interval()))]
        x=np.arange(len(fit))
        x=x*stf.get_sampling_interval()
        popt, pcov=curve_fit(self.func(), x, fit)"""


def deactivation_fit(self, start, protocol):
    """11580 for activation protocols, 4142 for deactivation protocols"""

    # need to check baseline between 8066 and 8076 for deactivation
    deact_df = pd.DataFrame(index=["a", "tau", "c"])
    plt.figure()
    max = np.ceil(np.sqrt(stf.get_size_channel()))
    index = 0
    for n in range(0, stf.get_size_channel(0)):
        trace = stf.get_trace(n)
        deact_start = np.argmax(trace[start:start + 75]) + start
        print deact_start
        baseline = np.mean(trace[8066:8076])
        x = 0
        fit = 0
        if "deact" in protocol:
            end = 0
            for z in range(deact_start, 8076):
                if np.allclose(trace[z], baseline, atol=1) is True:
                    end = z
                    print end
                    break
            fit = trace[int(deact_start):int(end)]
            x = np.arange(len(fit))
        else:
            fit = trace[int(deact_start):int(deact_start + (1 / stf.get_sampling_interval()))]
            x = np.arange(len(fit))

        x = x * stf.get_sampling_interval()
        popt = [0, 0, 0]

        popt, pcov = curve_fit(self.func, x, fit, maxfev=2000)
        plt.subplot(max, max, n + 1)
        plt.plot(x, fit)
        plt.plot(x, self.func(x, *popt))
        popt[1] = 1 / popt[1]
        print popt
        # except:
        #   popt=[0,0,0]
        #   print "couldn't fit"
```

```
            deact_df[str(index)] = popt
            index = index + 1
        plt.show()
        filename = stf.get_filename()[:-4]
        deact_df.to_csv(filename + ".csv")
        print deact_df
```

Appendix II -MultiAP3.py

```
import stf
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
from scipy.optimize import curve_fit
class AP(object):

#TO DO- make sure gets first AP
# add a smoothing function to AP_diff?

# create df that exports to csv containing info for whole cell at each stimulus

# add first AHP data to base list
# need a function for passive spiking before and after stimulus+extract AP from resting too

#TO DO-adaptation index not working-fixed
  # -numofAPmnot working- fixed
  # look at csv file and see if accurate with other software
  # delay evaluates as false even when all are true-should be fixed
  # no data for subthreshold traces strangely-hadnt added subthreshold function-fixed
  # select and deselect trace adding markers to start, AP peak, end, AHP peak
  # handle defunct action potentials if not screened out initially-maybe in AHP() add condition-done
  # stipulating that ten_percent_repol_time and ninety... have to be greater than 0 else pass this bit-luckily its at end-done
  # come up with best idea for getting baseline when cell is spontaneously active-e.g when dv/dt is <0.5-couldnt find a better soln
  # AHP peak detection is not accurate-use np.min and index of min instead of dv/dt-done
  # AP end could be more accurate, try a simple when data<threshold-done looks better
  # create new window so new markers are stored for each trace for quality control-done
  # peak deflection is off for some reason-find out-fixed
  # check subthreshold is functioning as not convinced-don't think selften_percent_max_time is working and membrane constant is weird
  # plot graphs as membrane constant in worked out to see if legit, do same for linear fit of depol and repol
  # do trial run of all stimuli and compare with hand values-done
  # look at stim_df_anal, changes to wave variable also changes df-issue lies in bad looping-fixed
  # look at AHP as time is not accurate to by hand-done
  # plot AHP and AP duration, AP amplitude and AP height for validation-done

  # amend absolute interval values with conversion from ms using sampling interval-dont need to do yet

  # convert all plots into ms
  # add a storage to core_df for average time constant capacitance and resistance calculation
  # work on time constant use fit cursors, slect fit with leastsq(1), get 2d numpy array containing fit values-
  # convert x to ms and ploty, vs logx then get gradient for tau -done

  # finish final plots-shouldbe done
  # change all absolute time values to relative-done
  # remove set_markers-done
  # add a AP number validator using threshold crossing-in some cases the last AP was not deteced-dont need to do
  # save all figures
  # save core_df to csv


  # core df needs to be destroyed after saving-done i think
  # missing first AP after higher stim-look at ends validation is not workingme thinks-fixed i think

  # Sort out how you handle ISI when only one AP is present-sorted i think
  # If starts equals 1 then end of AHP is estimated using the threshold of the first AP-modify, AHP, firstAP_data- sorted i think

  # Current issue locatedat stim anal-look at +30 csv stimdf to see where mean fails-done

   #Sometimes core analysis doesnt produce the right waveform and i have no idea why-problme in the mean?
   # Add function to define noise so rate of change threshold can be estimated for the user
   # its processing resting as if 0 is greater than 0 for some reason-move AP analysis, start and end from resting analysis
        """Tuples containing sweeps for certain stim condition"""
        resting=[]
        neg10=[]
        neg50=[]
        neg70=[]
        neg90=[]
        neg110=[]
        neg130=[]
        pos10=[]
        pos20=[]
        pos30=[]
        pos40=[]
        pos50=[]
        pos70=[]
        pos90=[]
        pos110=[]
        pos130=[]
        pos180=[]
        pos230=[]
        pos280=[]

        """Action potential features"""
        data=0
        array=0
        base_mean=0
        base_SD=0
        starts=[0,0]
        ends=[0,0]
```

```
thresholds=[0,0]
Potentials=0
AP_Absolute_Peaks=[0,0]
AP_Peak_Times=[0,0]
AP_Peak_Amplitudes=[0,0]
AP_Durations=[0,0]
AP_Depol_Times=[0,0]
AP_Repol_Times=[0,0]
AP_Max_Rises=[0,0]
AP_Max_Decays=[0,0]
AP_Rise_Decay_Ratio=[0,0]
first_AP_data=0
first_AP_array=0

"""Train features"""
AP_interspike_intervals=[0,0]
Av_ISI=0
First_ISI=0
ISI_Coeff_Vars=0
Num_Of_AP=0
Av_Firing_Freq=0
Check_Av_Firings=0
Instantaneous_Firing_Freq=0
Latencies=0
Adaptation_Indexes=0
Delay=0
Burst=0
Pause=0
Steady_Firing_Freq=0

"""AHP features"""
AHP_ends=[0,0]
AHP_Durations=[0,0]
AHP_Peak_Times=[0,0]
AHP_Peak_Amplitudes=[0,0]
AHP_Absolute_Peaks=[0,0]
AHP_Max_Rises=[0,0]
AHP_Max_Decays=[0,0]
"""Additional Allen Features"""
AHP_Fast_Troughs=[0,0]
AHP_Slow_Troughs=[0,0]
AHP_Slow_Trough_Times=[0,0]
AP_Heights=[0,0]
AP_Half_Height_Widths=[0,0]

"""Subthreshold features"""
Peak_Deflection=0
Peak_Deflection_Amplitude=0
Peak_Deflection_Time=0
Steady_Deflection=0
Steady_Deflection_Amplitude=0
Sag=0
ten_percent_max=0
ten_percent_max_time=0
ten_to_100=0
invert_ten_to_100=0
log_ten_to_100=0
membrane_constant=0
#update
#add analysis that looks at 2nd, 5th, 10th and 20th AP
index=["thresholds", "Potentials", "AP_Absolute_Peaks", "AP_Peak_Times", "AP_Peak_Amplitudes",
      "AP_Durations", "AP_Depol_Times", "AP_Repol_Times", "AP_Max_Rises", "AP_Max_Decays",
      "AP_Rise_Decay_Ratio", "first_AP_data", "first_AP_array", "AP_interspike_intervals",
      "Av_ISI", "First_ISI", "ISI_Coeff_Vars", "NumOfAP", "Av_Firing_Freq",
      "Check_Av_Firings", "Instantaneous_Firing_Freq", "Latencies",
      "Adaptation_Indexes", "Delay", "Burst", "Pause", "Steady_Firing_Freq",
      "AHP_Durations", "AHP_Peak_Times", "AHP_Peak_Amplitudes", "AHP_Absolute_Peaks",
      "AHP_Max_Rises", "AHP_Max_Decays", "AHP_Fast_Troughs", "AHP_Slow_Troughs",
      "AHP_Slow_Trough_Times", "AP_Heights", "AP_Half_Height_Widths", "Peak_Deflection",
      "Peak_Deflection_Amplitude", "Peak_Deflection_Time", "Steady_Deflection", "Steady_Deflection_Amplitude",
      "Sag", "ten_percent_max", "ten_percent_max_time", "membrane_constant", "thresholds_2", "AP_Peak_Amplitudes_2",
      "AP_Durations_2", "AP_Repol_Times_2", "AP_Max_Decays_2", "AHP_Durations_2", "AHP_Peak_Amplitudes_2", "AHP_Fast_Troughs_2",
      "AHP_Slow_Troughs_2", "AP_Half_Height_Widths_2", "thresholds_5", "AP_Peak_Amplitudes_5",
      "AP_Durations_5", "AP_Repol_Times_5", "AP_Max_Decays_5", "AHP_Durations_5", "AHP_Peak_Amplitudes_5", "AHP_Fast_Troughs_5",
      "AHP_Slow_Troughs_5", "AP_Half_Height_Widths_5", "thresholds_10", "AP_Peak_Amplitudes_10",
      "AP_Durations_10", "AP_Repol_Times_10", "AP_Max_Decays_10", "AHP_Durations_10", "AHP_Peak_Amplitudes_10", "AHP_Fast_Troughs_10",
      "AHP_Slow_Troughs_10", "AP_Half_Height_Widths_10", "thresholds_20", "AP_Peak_Amplitudes_20",
      "AP_Durations_20", "AP_Repol_Times_20", "AP_Max_Decays_20", "AHP_Durations_20", "AHP_Peak_Amplitudes_20", "AHP_Fast_Troughs_20",
      "AHP_Slow_Troughs_20", "AP_Half_Height_Widths_20"]

columns=["First Trace", "Second Trace", "Third Trace", "Average", "Std"]
core_columns=["-130", "-110", "-90", "-70", "-50", "-10", "0", "+10", "+20", "+30", "+40", "+50", "+70", "+90",
          "+110", "+130", "+180", "+230", "+280"]

stim_df=0

core_df = pd.DataFrame(index=index, columns=core_columns)


def core_df_refresh(self):

   self.core_df=pd.DataFrame(index=self.index, columns=self.core_columns)


def core_analysis(self, rheobase):
   # save core_df to csv
   # need to assess first whether any values are Nan
   # resistance plot
   plt.figure()
```

```
index=0
y=self.core_df.loc["Peak_Deflection_Amplitude", "-130":"-10"]
x=[]
y2=[]
for row in y:
    if str(row)!="nan":
        x.append(int(self.core_columns[index]))
        y2.append(row)
        index=index+1
    else:
        index=index+1
if len(x)>0:
    x=np.array(x)
    x2=x.astype(float)
     #to convert to mamps from pA
    resistance, intercept=np.polyfit(x2, y2, 1)
    print x
    print y2
    resistance=resistance*1000000000 #to give ohms
    self.core_df.loc["resistance, ohms", "0"]= resistance
    print resistance
    plt.subplot(331)
    #plt.title("Resistance Plot")
    plt.scatter(x, y2)
    plt.plot(x, y2)
    plt.xlabel("Current (pA)")
    plt.ylabel("Voltage (mV")

else:
    print "no subthreshold data"

# firing rate plot
index=5
x=[]
y2=[]
y=self.core_df.loc["Check_Av_Firings", "0":"+280"]
for row in y:
    if str(row)!="nan":
        x.append(int(self.core_columns[index]))
        y2.append(row)
        index=index+1
    else:
        index=index+1
if len(x) > 0:

    plt.subplot(333)
    #plt.title("Firing Rate")
    plt.scatter(x, y2)
    plt.xlabel("Current (pA)")
    plt.ylabel("No. Action Potentials")
else:
    print "no firing rate data"
# membrane time constant plot

index=0
x=[]
y2=[]
y=self.core_df.loc["membrane_constant", "-130":"-10"]
for row in y:
    if str(row)!="nan":
        x.append(int(self.core_columns[index]))
        y2.append(row)
        index=index+1
    else:
        index=index+1

if len(x) > 0:
    average_mem_constant = np.mean(y2)
    self.core_df.loc["average_mem_constant, ms", "0"]= average_mem_constant
    capacitance=(average_mem_constant/(self.core_df.loc["resistance, ohms", "0"]*1000))/0.0000314 # to give mOhms/tip area
    self.core_df.loc["capacitance mF", "0"]= capacitance
    plt.subplot(335)
    #plt.title("Membrane Time Constant")
    plt.scatter(x, y2)
    plt.xlabel("Current (pA)")
    plt.ylabel("Tau (ms)")
else:
    print "no membrane constant  data"

# phase plot of first AP"
x=self.core_df.loc["first_AP_data", rheobase]
y=self.core_df.loc["first_AP_array", rheobase]
if len(x)>0:
    x=x[2:]
    y=y[2:]

    plt.subplot(337)
    #plt.title("Action Potential Phase Plot")
    plt.plot(x, y)
    plt.xlabel("Voltage (mV)")
    plt.ylabel("dv/dt")
    # first AP plot
    y=x
    x=list(range(len(y)))
    x = np.array(x)
    x = x * 0.2
    plt.subplot(339)
    #plt.title("Action Potential")
```

```
            plt.plot(x, y)
            plt.xlabel("Time (ms)")
            plt.ylabel("Voltage (mV)")
        figname=stf.get_filename()[0:-4]+".png"
        plt.savefig(figname)
        plt.show()
        # example graphs
        # average

        outfile=stf.get_filename()[0:-4]+".csv"
        self.core_df.to_csv(outfile)
        self.core_df=pd.DataFrame(index=self.index, columns=self.core_columns)

    def stim_df_anal(self, condition):
        #get mean and std
        traces = 0
        wave = 0
        new_wave=0
        new_new_wave=0
        for row in self.stim_df.itertuples():
            row_index = row[0]
            if row[0]=="Delay"or row[0]=="Burst" or row[0]=="Pause":
                true_count=row[1:4].count(True)
                false_count=row[1:4].count(False)
                if true_count>false_count:
                    self.stim_df.set_value(row_index, "Average", True)
                else:
                    self.stim_df.set_value(row_index, "Average", False)
            elif row[0] == "first_AP_data" or row[0] == "first_AP_array":
                #TO DO-figure out how to average traces -maybe plot values and plot fit to it or align peaks, and cut
                # for value rangemax length: if
                # find middle peak, find distance other peaks are from, move all points this distance,
                # trim from the end to smallest
                # averagenp arrays
                if int(condition)>0 and sum(self.starts)>0:


                    if row[0]=="first_AP_data":
                        wave = self.stim_df.loc["first_AP_data"]
                    elif row[0]=="first_AP_array":
                        wave = self.stim_df.loc["first_AP_array"]
                    depol_times = self.stim_df.loc["AP_Depol_Times"] / stf.get_sampling_interval()
                    new_wave = wave

                    median = np.median(depol_times[0:3])


                    smallest=len(new_wave[0])
                    dist_of_smallest=0
                    dist=[]
                    roll=[]
                    length=[]
                    for n in range(0,3):
                        dist.append(median - depol_times[n])
                        roll.append(len(new_wave[n]) + int(dist[n]))
                        length.append(len(new_wave[n]))
                        #if len(new_wave[n])<smallest:
                         #   smallest=len(new_wave[n])
                          #  dist_of_smallest=dist[n]
                    smallest=np.min(length)
                    smallest_index=np.argmin(length)
                    dist_of_smallest=dist[smallest_index]



                    for n in range(0, 3):
                        new_new_wave=np.roll(new_wave[n], roll[n])
                        if dist_of_smallest<0 or dist_of_smallest==0: #edited 231117-does it work? No
                            traces=np.empty([3, smallest])
                            traces[n]=new_new_wave[0:smallest]
                        else:
                            trace_length=len(new_new_wave[int(dist_of_smallest):smallest])
                            traces=np.empty([3, trace_length]) #error with file 17o16000 can not broadcast input fromshape 517 into 516- what is the problem here?? maybe create an exception to handle it, or maybe just increase by 1


                            traces[n]=new_new_wave[int(dist_of_smallest):smallest]

                    average_trace=sum(traces)/3
                    squared_deviations=[]
                    for n in traces:
                        squared_deviations.append((n-average_trace)**2)
                    stdOftrace=sum(squared_deviations)/3-1
                    #stdOftrace=((traces[0]-average_trace)**2+(traces[1]-average_trace)**2+(traces[2]-average_trace)**2)/(3-1)


                    self.stim_df.set_value(row_index, "Average", average_trace) #might not work
                    self.stim_df.set_value(row_index, "Std", stdOftrace)
                else:
                    self.stim_df.set_value(row_index, "Average", 0)
                    self.stim_df.set_value(row_index, "Std", 0)
            else:
                row = np.array(row[1:4])
                row_mean = np.mean(row)
                row_std = np.std(row)
                self.stim_df.set_value(row_index, "Average", row_mean)
                self.stim_df.set_value(row_index, "Std", row_std)
```

```
    self.core_df[condition]=self.stim_df["Average"]
    output_file=stf.get_filename()[0:-4]+str(condition)+".csv"
    self.stim_df.to_csv(output_file)


def sweep_select(self):
    self.resting = raw_input("0").split(",")
    self.neg10 = raw_input("-10").split(",")
    self.neg50 = raw_input("-50").split(",")
    self.neg70 = raw_input("-70").split(",")
    self.neg90 = raw_input("-90").split(",")
    self.neg110 = raw_input("-110").split(",")
    self.neg130 = raw_input("-130").split(",")
    self.pos10 = raw_input("+10").split(",")
    self.pos20 = raw_input("+20").split(",")
    self.pos30 = raw_input("+30").split(",")
    self.pos40 = raw_input("+40").split(",")
    self.pos50 = raw_input("+50").split(",")
    self.pos70 = raw_input("+70").split(",")
    self.pos90 = raw_input("+90").split(",")
    self.pos110 = raw_input("+110").split(",")
    self.pos130 = raw_input("+130").split(",")
    self.pos180 = raw_input("+180").split(",")
    self.pos230 = raw_input("+230").split(",")
    self.pos280 = raw_input("+280").split(",")


"""Main function-calls newly assigned tuples"""

def cell(self, thresh, start):
    """a value of 2.0 is a good estimate for threshold rate of change"""
    self.core_df_refresh()
    self.analysis(self.resting, "0", thresh, start)
    self.analysis(self.neg10, "-10", thresh, start)
    self.analysis(self.neg50, "-50", thresh, start)
    self.analysis(self.neg70, "-70", thresh, start)
    self.analysis(self.neg90, "-90", thresh, start)
    self.analysis(self.neg110, "-110", thresh, start)
    self.analysis(self.neg130, "-130", thresh, start)

    self.analysis(self.pos10, "+10", thresh, start)
    self.analysis(self.pos20, "+20", thresh, start)
    self.analysis(self.pos30, "+30", thresh, start)
    self.analysis(self.pos40, "+40", thresh, start)
    self.analysis(self.pos50, "+50", thresh, start)
    self.analysis(self.pos70, "+70", thresh, start)
    self.analysis(self.pos90, "+90", thresh, start)
    self.analysis(self.pos110, "+110", thresh, start)
    self.analysis(self.pos130, "+130", thresh, start)
    self.analysis(self.pos180, "+180", thresh, start)
    self.analysis(self.pos230, "+230", thresh, start)
    self.analysis(self.pos280, "+280", thresh, start)

"""To call all functions"""
def analysis(self, listnm, condition, thresh, start):
    trace_num = 0
    self.stim_df = pd.DataFrame(index=self.index, columns=self.columns)
    condition=condition
    if len(listnm)>1:
        plt.figure()
        for trace in listnm:
            self.refresh_x()
            trace=int(trace)
            self.AP_diff(trace)
            self.baseline()
            if int(condition)>0:
                # add a refresh here just write them out all subthreshold measurements
                self.Peak_Deflection=0
                self.Peak_Deflection_Amplitude=0
                self.Peak_Deflection_Time=0
                self.Steady_Deflection=0
                self.Steady_Deflection_Amplitude=0
                self.Sag=0
                self.ten_percent_max=0
                self.ten_percent_max_time=0
                self.membrane_constant=0

                self.AP_start(thresh, start)
                self.AP_end()
                self.AHP_end()
                self.AP_analysis()
                self.AHP()

                x = list(range(0, stf.get_size_trace(trace)))
                plt.subplot(2, 2, trace_num + 1)
                plt.plot(x, self.data)
                if sum(self.starts)>0:
                    plt.scatter(self.starts, self.thresholds)
                    plt.scatter(self.ends, self.thresholds)
                    plt.scatter(self.AP_Peak_Times, self.AP_Absolute_Peaks)
                    plt.scatter(self.AHP_Peak_Times, self.AHP_Absolute_Peaks)
                    plt.plot([1, (stf.get_size_trace()-1)], [self.Potentials, self.Potentials])
                    for n in self.starts[:-1]:
                        index = self.starts.index(n)
                        # plt.plot([n, self.ends[index]], [self.data[n], self.data[self.ends[index]]])

                        # plt.plot([self.AHP_Peak_Times[index], self.AHP_Peak_Times[index]],
                        # [self.AHP_Absolute_Peaks[index], self.AP_Absolute_Peaks[index]])
```

```
                    plt.plot([self.ends[index], self.AHP_ends[index]],
                        [self.thresholds[index], self.thresholds[index]])


            elif int(condition)<0:
                self.subthreshold()
                x = list(range(0, stf.get_size_trace(trace)))
                plt.subplot(2, 2, trace_num+1)
                plt.plot(x, self.data)  #peak deflections, sag,, steadystate
                plt.scatter(self.Peak_Deflection_Time, self.Peak_Deflection)
                plt.plot([1,stf.get_size_trace()-1], [self.Steady_Deflection, self.Steady_Deflection])
                plt.plot([1, stf.get_size_trace()-1], [self.Potentials, self.Potentials])
                #plt.plot(list(range(len(self.ten_to_100))), self.ten_to_100)
                #plt.plot(list(range(len(self.log_ten_to_100))), self.log_ten_to_100)

            elif int(condition)==0:
                x=list(range(0, stf.get_size_trace()))
                plt.subplot(2,2, trace_num+1)
                plt.plot(x, self.data)
                plt.plot([1, stf.get_size_trace()-1], [self.Potentials, self.Potentials])


            if self.Num_Of_AP>2:
                second_AP=[self.thresholds[1], self.AP_Peak_Amplitudes[1], self.AP_Durations[1], self.AP_Repol_Times[1],
                        self.AP_Max_Decays[1], self.AHP_Durations[1], self.AHP_Peak_Amplitudes[1], self.AHP_Fast_Troughs[1],
                        self.AHP_Slow_Troughs[1], self.AP_Half_Height_Widths[1]]
            else:
                second_AP=[0,0,0,0,0,0,0,0,0,0]

            if self.Num_Of_AP>5:
                fourth_AP=[self.thresholds[4], self.AP_Peak_Amplitudes[4], self.AP_Durations[4], self.AP_Repol_Times[4],
                        self.AP_Max_Decays[4], self.AHP_Durations[4], self.AHP_Peak_Amplitudes[4], self.AHP_Fast_Troughs[4],
                        self.AHP_Slow_Troughs[4], self.AP_Half_Height_Widths[4]]
            else:
                fourth_AP=[0,0,0,0,0,0,0,0,0,0]

            if self.Num_Of_AP>10:
                tenth_AP=[self.thresholds[9], self.AP_Peak_Amplitudes[9], self.AP_Durations[9], self.AP_Repol_Times[9],
                        self.AP_Max_Decays[9], self.AHP_Durations[9], self.AHP_Peak_Amplitudes[9], self.AHP_Fast_Troughs[9],
                        self.AHP_Slow_Troughs[9], self.AP_Half_Height_Widths[9]]
            else:
                tenth_AP=[0,0,0,0,0,0,0,0,0,0]

            if self.Num_Of_AP>20:
                twenty_AP=[self.thresholds[19], self.AP_Peak_Amplitudes[19], self.AP_Durations[19], self.AP_Repol_Times[19],
                        self.AP_Max_Decays[19], self.AHP_Durations[19], self.AHP_Peak_Amplitudes[19], self.AHP_Fast_Troughs[19],
                        self.AHP_Slow_Troughs[19], self.AP_Half_Height_Widths[19]]
            else:
                twenty_AP=[0,0,0,0,0,0,0,0,0,0]



            #if subthreshold trace call def subthreshold function
            #add extras here
            #may have issue where lists are only one value just maybr just use variables
            x=[self.thresholds[0], self.Potentials, self.AP_Absolute_Peaks[0], self.AP_Peak_Times[0], self.AP_Peak_Amplitudes[0],
                self.AP_Durations[0], self.AP_Depol_Times[0], self.AP_Repol_Times[0], self.AP_Max_Rises[0], self.AP_Max_Decays[0],
                self.AP_Rise_Decay_Ratio[0], self.first_AP_data, self.first_AP_array, self.AP_interspike_intervals[0],
                self.Av_ISI, self.First_ISI, self.ISI_Coeff_Vars, self.Num_Of_AP, self.Av_Firing_Freq,
                self.Check_Av_Firings, self.Instantaneous_Firing_Freq, self.Latencies,
                self.Adaptation_Indexes, self.Delay, self.Burst, self.Pause, self.Steady_Firing_Freq,
                self.AHP_Durations[0], self.AHP_Peak_Times[0], self.AHP_Peak_Amplitudes[0], self.AHP_Absolute_Peaks[0],
                self.AHP_Max_Rises[0], self.AHP_Max_Decays[0], self.AHP_Fast_Troughs[0], self.AHP_Slow_Troughs[0],
                self.AHP_Slow_Trough_Times[0], self.AP_Heights[0], self.AP_Half_Height_Widths[0], self.Peak_Deflection,
                self.Peak_Deflection_Amplitude, self.Peak_Deflection_Time, self.Steady_Deflection,
                self.Steady_Deflection_Amplitude, self.Sag, self.ten_percent_max,
                self.ten_percent_max_time, self.membrane_constant]
            x=x+second_AP+fourth_AP+tenth_AP+twenty_AP

            self.stim_df[self.columns[trace_num]]=x
            trace_num=trace_num+1


        self.stim_df_anal(condition)
        figname=stf.get_filename()[0:-4]+condition+".png"
        plt.savefig(figname)
        plt.show()

    else:
        pass

    def refresh_x(self):

        """lists=[self.thresholds, self.AP_Absolute_Peaks, self.AP_Peak_Times, self.AP_Peak_Amplitudes,
                self.AP_Durations, self.AP_Depol_Times, self.AP_Repol_Times, self.AP_Max_Rises, self.AP_Max_Decays,
                self.AP_Rise_Decay_Ratio, self.AP_interspike_intervals, self.AHP_Durations, self.AHP_Peak_Times, self.AHP_Peak_Amplitudes,
self.AHP_Absolute_Peaks,
                self.AHP_Max_Rises, self.AHP_Max_Decays, self.AHP_Fast_Troughs, self.AHP_Slow_Troughs,
                self.AHP_Slow_Trough_Times, self.AP_Heights, self.AP_Half_Height_Widths]

        variables=[self.Potentials, self.Av_ISI, self.First_ISI, self.ISI_Coeff_Vars, self.Num_Of_AP, self.Av_Firing_Freq,
                self.Check_Av_Firings, self.Instantaneous_Firing_Freq, self.Latencies,
                self.Adaptation_Indexes, self.Delay, self.Burst, self.Pause, self.Steady_Firing_Freq, self.Peak_Deflection,
                self.Peak_Deflection_Amplitude, self.Peak_Deflection_Time, self.Steady_Deflection,
                self.Steady_Deflection_Amplitude, self.Sag, self.ten_percent_max,
                self.ten_percent_max_time, self.membrane_constant]"""

        self.first_AP_data=0
```

```
        self.first_AP_array=0
        self.thresholds=[0,0]
        self.AP_Absolute_Peaks=[0,0]
        self.AP_Peak_Times=[0,0]
        self.AP_Peak_Amplitudes=[0,0]
        self.AP_Durations=[0,0]
        self.AP_Depol_Times=[0,0]
        self.AP_Repol_Times=[0,0]
        self.AP_Max_Rises=[0,0]
        self.AP_Max_Decays=[0,0]
        self.AP_Rise_Decay_Ratio=[0,0]
        self.AP_interspike_intervals=[0,0]
        self.AHP_Durations=[0,0]
        self.AHP_Peak_Times=[0,0]
        self.AHP_Peak_Amplitudes=[0,0]
        self.AHP_Absolute_Peaks=[0,0]
        self.AHP_Max_Rises=[0,0]
        self.AHP_Max_Decays=[0,0]
        self.AHP_Fast_Troughs=[0,0]
        self.AHP_Slow_Troughs=[0,0]
        self.AHP_Slow_Trough_Times=[0,0]
        self.AP_Heights=[0,0]
        self.AP_Half_Height_Widths=[0,0]

        self.Potentials=0
        self.Av_ISI=0
        self.First_ISI=0
        self.ISI_Coeff_Vars=0
        self.Num_Of_AP=0
        self.Av_Firing_Freq=0
        self.Check_Av_Firings=0
        self.Instantaneous_Firing_Freq=0
        self.Latencies=0
        self.Adaptation_Indexes=0
        self.Delay=0
        self.Burst=0
        self.Pause=0
        self.Steady_Firing_Freq=0
        self.Peak_Deflection=0
        self.Peak_Deflection_Amplitude=0
        self.Peak_Deflection_Time=0
        self.Steady_Deflection=0
        self.Steady_Deflection_Amplitude=0
        self.Sag=0
        self.ten_percent_max=0
        self.ten_percent_max_time=0
        self.membrane_constant=0
        """for n in lists:
           index=lists.index(n)
           lists[index] =[0,0]

        for n in variables:
           index=variables.index(n)
           variables[index]= 0  #this probably isnt working-doesnt actually reset the variable
        print self.Peak_Deflection
        print self.Sag"""



    def AP_diff(self, trace):
        """ Puts trace in variable data, creates differentiated array, defines baseline mean and SD in differentiated array"""
        self.data=stf.get_trace(trace)

        self.array=np.empty(stf.get_size_trace())
        for n in range(0, stf.get_size_trace()-1):
           self.array[n]= ((self.data[n+1]-self.data[n])/stf.get_sampling_interval())
        # define baseline mean, SD,
        #self.array=gaussian_filter1d(self.array, 2)

        self.base_mean = self.array[int(40/stf.get_sampling_interval()):int(140/stf.get_sampling_interval())].mean() ##this needs to be modified so doesnt include any spontaneous AP

        self.base_SD = self.array[int(40/stf.get_sampling_interval()):int(140.2/stf.get_sampling_interval())].std()

    def baseline(self): #make better-to avoid any spontaneous APs-might have to take this into account
        self.Potentials=0

        baseline=np.mean(self.data[0:int(220/stf.get_sampling_interval())])

        self.Potentials=baseline

    """Sets AP start points"""
    def AP_start(self, thresh, start): # edit to take no arguments, update start and end as hard numbers as they are consistent
        n=start #make conversion

        self.starts=[]
        while n < int(1218/stf.get_sampling_interval()):  # amek conversion
           if self.array[n]>thresh:   #self.base_mean+(self.base_SD*4)

              self.starts.append(n-1)   #ajdusted to make start at n-1
              n=n+int(10/stf.get_sampling_interval())
           else:
              n=n+1



    """Loop through starts list to create ends list"""
    def AP_end(self):
        self.thresholds=[]
```

```
        self.ends=[]
        if sum(self.starts)>0:
            for n in self.starts: #could just say when data is next lower than threshold that is end
                value=n+int(2.2/stf.get_sampling_interval())
                index=self.starts.index(n)
                threshold=self.data[n]
                self.thresholds.append(threshold)
                while value>n+int(2/stf.get_sampling_interval()) and value<n+int(100/stf.get_sampling_interval()):
                    if self.data[value]<self.data[n+1]:
                        self.ends.append(value)

                        value=value+1
                        break
                    else:
                        value=value+1
            #check for accurate first AP
            if len(self.starts)!=len(self.ends):
                # delete irst flase AP caused by stim on
                # and
                # use threshold of second AP to findnew start



                n=self.starts[0]
                value=self.starts[0]
                while value<n+int(100/stf.get_sampling_interval()):
                    if self.data[value]>self.thresholds[1]:
                        self.starts[0]=value
                        self.thresholds[0]=self.data[value-1]

                        break
                    else:
                        value=value+1
                 # to get end
                n=self.starts[0]
                value=self.starts[0]+int(2.2/stf.get_sampling_interval())
                while value>n+int(2/stf.get_sampling_interval()) and value<n+int(100/stf.get_sampling_interval()):
                    if self.data[value]<self.data[n+1]:
                        self.ends.append(value)
                        self.ends=np.roll(np.array(self.ends), 1)
                        self.ends=self.ends.tolist()

                        break
                    else:
                        value=value+1

                if self.ends[0]-self.starts[0]>50/stf.get_sampling_interval():

                    del self.starts[0]
                    del self.thresholds[0]


        else:
            print "no APs detected"
            self.thresholds=[0,0,0]
            self.ends=[0,0,0]

    def AHP_end(self):
        self.AHP_ends = []
        if sum(self.starts) > 0:

            for n in self.ends[0:(len(self.ends))]:  # change
                index = self.ends.index(n)


                if n == self.ends[-1]:
                    for x in range(int(n + 15 / stf.get_sampling_interval()),
                                   int(n + 500 / stf.get_sampling_interval())):
                        if np.allclose(self.data[x], self.data[self.starts[0]], atol=6) is True:    ##Just fixed poss problem atol too low thereore no AHP end appended
                            self.AHP_ends.append(x)
                            break
                else:
                    for x in range(int(n+15/stf.get_sampling_interval()),
                                   int(n+500/stf.get_sampling_interval())):
                        if np.allclose(self.data[x], self.data[n], atol=3) is True:
                            self.AHP_ends.append(x)
                            break

            for n in self.AHP_ends[:-1]:
                index=self.AHP_ends.index(n)
                if n>self.starts[index+1]:
                    self.AHP_ends[index]=self.starts[index+1]

        #need to add an assurance if AHP not defined before next AP start


    """Set and get AP parameters"""
    def AP_analysis(self):
            self.AP_Absolute_Peaks=[]
            self.AP_Peak_Times=[]
            self.AP_Peak_Amplitudes=[]
            self.AP_Durations=[]
            self.AP_Depol_Times=[]
            self.AP_Repol_Times=[]
            self.AP_Max_Rises=[]
            self.AP_Max_Decays=[]
            self.AP_Rise_Decay_Ratio=[]
            self.AP_interspike_intervals=[]
            self.Av_Firing_Freq=0
```

```
self.Steady_Firing_Freq=0
self.Instantaneous_Firing_Freq=0
self.Num_Of_AP=0
self.ISI_Coeff_Vars=0
self.Av_ISI=0
self.First_ISI=0
self.Check_Av_Firings=0
self.Latencies=0
self.Adaptation_Indexes=0
self.Delay=0
self.Burst=0
self.Pause=0

if sum(self.starts)>0:
    for n in self.starts:
        index=self.starts.index(n)

        # AP Peak Time
        max_value=np.max(self.data[self.starts[index]:self.ends[index]])
        for value in range(int(self.starts[index]), int(self.ends[index])):
          if self.data[value]==max_value:
            AP_Peak_T=value

            self.AP_Peak_Times.append(AP_Peak_T)
            AP_abs_peak=self.data[value]
            self.AP_Absolute_Peaks.append(AP_abs_peak)
            AP_amplitude = self.data[value] -self.thresholds[index]
            self.AP_Peak_Amplitudes.append(AP_amplitude)
            break

        AP_duration=(self.ends[index]-self.starts[index])*stf.get_sampling_interval()
        self.AP_Durations.append(AP_duration)

        AP_depol_T=(AP_Peak_T-self.starts[index])*stf.get_sampling_interval()
        self.AP_Depol_Times.append(AP_depol_T)

        AP_repol_T=(self.ends[index]-AP_Peak_T)*stf.get_sampling_interval()
        self.AP_Repol_Times.append(AP_repol_T)

        AP_max_rise=np.max(self.array[self.starts[index]:self.ends[index]])
        self.AP_Max_Rises.append(AP_max_rise)

        AP_max_decay=np.min(self.array[self.starts[index]:self.ends[index]])
        self.AP_Max_Decays.append(AP_max_decay)

        rise_decay_ratio=self.AP_Max_Rises[index]/self.AP_Max_Decays[index]
        self.AP_Rise_Decay_Ratio.append(rise_decay_ratio)

    self.Num_Of_AP = len(self.starts)

    # Interspike interval

    for n in self.AP_Peak_Times[0:(len(self.AP_Peak_Times) - 1)]:
      index = self.AP_Peak_Times.index(n)
      interspike_interval = (self.AP_Peak_Times[index + 1] - self.AP_Peak_Times[
        index]) * stf.get_sampling_interval()
      self.AP_interspike_intervals.append(interspike_interval)
    if sum(self.AP_interspike_intervals)>0:
      # First ISI
      self.First_ISI = self.AP_interspike_intervals[0]
      # Instantaneous Firing Freq
      # Interval between first two peaks
      self.Instantaneous_Firing_Freq = (1 / self.AP_interspike_intervals[0]) * 1000
      # Steady Firing Rate
      # Average of intervals between last 5 peaks
      av_last5_ISI = sum(self.AP_interspike_intervals[-5:]) / len(
        self.AP_interspike_intervals[-5:])
      self.Steady_Firing_Freq = (1 / av_last5_ISI) * 1000
      # Average Firing Freq
      av_ISI = sum(self.AP_interspike_intervals) / len(self.AP_interspike_intervals)
      self.Av_ISI = av_ISI  ###
      self.Av_Firing_Freq = (1 / av_ISI) * 1000  ## or divide number of AP by time of stim
      # ISI coefficient of variation-std/av ISI
      self.ISI_Coeff_Vars = (np.std(self.AP_interspike_intervals)) / av_ISI  ###
      # Delay- if time to first AP >av ISI
      if ((self.AP_Peak_Times[0] * stf.get_sampling_interval()) - 220) > self.Av_ISI:  # this could be wrong
        self.Delay = "True"
      else:
        self.Delay = "False"

    else:
      self.First_ISI=0
      self.Instantaneous_Firing_Freq=0
      self.Steady_Firing_Freq=0
      self.Av_ISI=0
      self.Av_Firing_Freq=0
      self.ISI_Coeff_Vars=0
      self.Delay="False"
      self.AP_interspike_intervals = [0]



    check_av_firing = len(self.starts) / 1
    self.Check_Av_Firings = check_av_firing  ###
    # Save first AP array and data slice
    # need array and data slices from action potentials
    if sum(self.AHP_ends)>0:
      self.first_AP_data = self.data[self.starts[0]:self.AHP_ends[0]]
      self.first_AP_array = self.array[self.starts[0]:self.AHP_ends[0]]
```

```
            else:
                self.first_AP_array=[0,0,0]
                self.first_AP_array=[0,0,0]

            # Latency-from onset of stim to first spike

            self.Latencies = (self.AP_Peak_Times[0] * stf.get_sampling_interval()) - 220### this may be wrong


            # Adaptation index-rate at which speeds up or slow down
            if len(self.AP_interspike_intervals) > 1:
                answers = []

                for n in range(0, len(self.AP_interspike_intervals)-1):
                    numerator = self.AP_interspike_intervals[n + 1] - self.AP_interspike_intervals[n]
                    denominator = self.AP_interspike_intervals[n + 1] + self.AP_interspike_intervals[n]
                    answer = numerator / denominator
                    answers.append(answer)

                self.Adaptation_Indexes = (1 / float((len(self.AP_interspike_intervals) - 1))) * sum(answers)  ###



            # Burst-if first two ISI< or equal to 5ms

                if self.AP_interspike_intervals[0] and self.AP_interspike_intervals[1] <= 0.005:
                    self.Burst = "True"
                else:
                    self.Burst = "False"

            # Pause-i any ISI < 3X duration ISI BEFORE AND AFTER
                for n in self.AP_interspike_intervals[1:-1]:
                    index=self.AP_interspike_intervals.index(n)
                    if int(n) > 3 * self.AP_interspike_intervals[index - 1] and n > 3 * self.AP_interspike_intervals[index + 1]:
                        self.Pause = "True"
                    else:
                        self.Pause = "False"

        else:

            self.AP_Absolute_Peaks=[0,0,0]
            self.AP_Peak_Times = [0,0,0]
            self.AP_Peak_Amplitudes = [0,0,0]
            self.AP_Durations = [0,0,0]
            self.AP_Depol_Times = [0,0,0]
            self.AP_Repol_Times = [0,0,0]
            self.AP_Max_Rises = [0,0,0]
            self.AP_Max_Decays = [0,0,0]
            self.AP_Rise_Decay_Ratio = [0,0,0]
            self.AP_interspike_intervals = [0,0,0]
            self.first_AP_data=np.array([0,0,0])
            self.first_AP_array=np.array([0,0,0])


    """Subthreshold Analysis"""
    """ calculate peak deflection, sag, steady deflection, fit exp curve between 10% and max, average
    time constants of these to find membrane time constant, plot voltage response against current step to get
    resistance from curve. """
    def subthreshold(self):
            self.Peak_Deflection=0
            self.Peak_Deflection_Amplitude=0
            self.Steady_Deflection=0
            self.Steady_Deflection_Amplitude=0
            self.Sag=0
            self.Peak_Deflection_Time=0
            self.ten_percent_max_time = 0
            self.ten_to_100=0
            self.invert_ten_to_100=0
            self.log_ten_to_100=0
            self.membrane_constant=0

            # peak deflection
            stim_on=int(220/stf.get_sampling_interval()) ##
            stim_off=int(1200/stf.get_sampling_interval())

            self.Peak_Deflection=self.data[stim_on:stim_off].min()
            self.Peak_Deflection_Amplitude=self.Peak_Deflection-self.Potentials

            self.Peak_Deflection_Time=stim_on+self.data[stim_on:stim_off].argmin()
            # steady deflection
            self.Steady_Deflection=self.data[int(1000/stf.get_sampling_interval()):stim_off].mean()        ##
            self.Steady_Deflection_Amplitude=self.Steady_Deflection-self.Potentials
            # sag
            self.Sag=self.Steady_Deflection-self.Peak_Deflection


            #10% of max
            self.ten_percent_max=self.Peak_Deflection-(self.Peak_Deflection_Amplitude*0.9)


            for value in range(int(220/stf.get_sampling_interval()), int(400/stf.get_sampling_interval())):                    ##
                if self.data[value]<self.ten_percent_max:
                    self.ten_percent_max_time=value
                    break

            self.ten_to_100=self.data[int(self.ten_percent_max_time):int(self.Peak_Deflection_Time)]
            x=list(range(len(self.ten_to_100)))
            y=self.ten_to_100
            min_y=np.min(y)
```

```
        max_y=np.max(y)
        x=np.array(x)
        x=x*stf.get_sampling_interval()
        y=y-min_y
        est_para=(max_y, 0.1, 1)
        popt, pcov= curve_fit(self.func, x, y, p0=est_para, maxfev=20000)
        y2=self.func(x, *popt)
        tauV=y2[0]*0.37
        tau=0
        y3=y2.tolist()
        for n in y3:
            index=y3.index(n)
            if n<tauV:
                tau=x[index]
                break
        self.membrane_constant=tau


def func(self, x, a, b, c):
    return a*np.exp(-b*x)+c




"""AHP analysis

analysing time between end of an AP and start of next AP
loop through end times
index end time
use starts[index+1] to get end of AHP- maybe end AHP a couple of data points earlier than next AP"""
def AHP(self):
        self.AHP_Durations=[]
        self.AHP_Peak_Times=[]
        self.AHP_Peak_Amplitudes=[]
        self.AHP_Max_Rises=[]
        self.AHP_Max_Decays=[]
        self.AHP_Fast_Troughs=[]
        self.AHP_Slow_Troughs=[]
        self.AHP_Slow_Trough_Times=[]
        self.AHP_Absolute_Peaks=[]
        self.AP_Heights=[]
        self.AP_Half_Height_Widths=[]
        ###TO DO- this isn't collecting all AHPs-why??- cos it said -2, -1 should collect first 4
        if sum(self.AHP_ends)>0:


            for n in self.AHP_ends:#change

                index=self.AHP_ends.index(n)

                #AHP Duration

                AHP_duration=(n-self.ends[index])*stf.get_sampling_interval()
                self.AHP_Durations.append(AHP_duration)


                #AHP_Peak-CHANGE ALL OF THIS-done
                min=np.min(self.data[self.ends[index]:self.AHP_ends[index]])
                min_index=np.argmin(self.data[self.ends[index]:n])+self.ends[index]
                self.AHP_Peak_Times.append(min_index)
                self.AHP_Absolute_Peaks.append(min)
                AHP_peak_amplitude=self.thresholds[index]-min
                self.AHP_Peak_Amplitudes.append(AHP_peak_amplitude)


                #AHP Max Rise
                AHP_max_rise=self.array[int(self.ends[index]):int(self.AHP_ends[index])].max()
                self.AHP_Max_Rises.append(AHP_max_rise)
                # AHP max decay
                AHP_max_decay = self.array[int(self.ends[index]):int(self.AHP_ends[index])].min()
                self.AHP_Max_Decays.append(AHP_max_decay)

                #AHP fast and slow trough

                AHP_fast_trough_start=self.AP_Peak_Times[index]
                AHP_slow_trough_start = self.AP_Peak_Times[index]+(5/stf.get_sampling_interval())
                AHP_slow_trough_min=self.data[int(AHP_slow_trough_start):self.AHP_ends[index]].min()
                AHP_fast_trough_min=self.data[int(AHP_fast_trough_start):int(AHP_slow_trough_start)].min()
                AHP_slow_trough_time=(((AHP_slow_trough_start-AHP_fast_trough_start)/
                            (self.AHP_ends[index]-AHP_fast_trough_start))*stf.get_sampling_interval())
                self.AHP_Slow_Trough_Times.append(AHP_slow_trough_time)
                self.AHP_Slow_Troughs.append(AHP_slow_trough_min)
                self.AHP_Fast_Troughs.append(AHP_fast_trough_min)

                # AP_height (AP_abs-AHP_abs)
                AP_height=self.AP_Absolute_Peaks[index]-self.AHP_Absolute_Peaks[index]
                self.AP_Heights.append(AP_height)


                # AP Width at half height
                # if v insensitive maybe use wider net and pick median
                # very inaccurate, plot a linear line through 10-90%
                # depol 10-90
                ten_percent_depol=self.AP_Absolute_Peaks[index]-(self.AP_Peak_Amplitudes[index]*0.9)
                ninety_percent_depol=self.AP_Absolute_Peaks[index]-(self.AP_Peak_Amplitudes[index]*0.1)
```

```
            ten_percent_depol_time=0
            ninety_percent_depol_time=0
            ten_percent_repol_time=0
            ninety_percent_repol_time=0
            """depol 10% and 90% times"""
            for value in range(self.starts[index], self.AP_Peak_Times[index]):
              if self.data[value]>ten_percent_depol:
                  ten_percent_depol_time=value
                  break

            for value in range(self.starts[index], self.AP_Peak_Times[index]):
              if self.data[value]>ninety_percent_depol:
                  ninety_percent_depol_time=value
                  break
            """repol 10% and 90% times"""
            for value in range(self.AP_Peak_Times[index], self.ends[index]):
              if self.data[value]<ten_percent_depol:
                  ten_percent_repol_time=value
                  break

            for value in range(self.AP_Peak_Times[index], self.ends[index]):
              if self.data[value]<ninety_percent_depol:
                  ninety_percent_repol_time=value
                  break

            print ninety_percent_repol_time
            print ten_percent_repol_time

            half_AP_height = self.AP_Absolute_Peaks[index] - (self.AP_Heights[index] / 2)

            """depol"""
            x=[]
            for n in range(0, ninety_percent_depol_time-ten_percent_depol_time):
              x.append(n)
            if x==[]:
              print "x is empty-defunct AP may be present"
              pass
            else:
              y=self.data[ten_percent_depol_time:ninety_percent_depol_time]
              line=np.polyfit(x, y, 1)
              gradient=line[0]
              y_intercept=line[1]

              first_cursor=((half_AP_height-y_intercept)/gradient)+ten_percent_depol_time
              print first_cursor
              """repol"""

              x2 = []
              for n in range(0, (ten_percent_repol_time-ninety_percent_repol_time)):
                x2.append(n)

              print x2
              if x2==[]:
                print "x2 is empty-defunct AP present"
                pass
              else:
                y2 = self.data[ninety_percent_repol_time:ten_percent_repol_time]
                line = np.polyfit(x2, y2, 1)
                gradient = line[0]
                y_intercept = line[1]

                second_cursor=(half_AP_height-y_intercept)/gradient+ninety_percent_repol_time
                print second_cursor
                AP_half_height_width = (second_cursor - first_cursor) * stf.get_sampling_interval()
                self.AP_Half_Height_Widths.append(AP_half_height_width)

              #any issue here means that there is a defunct action potential
              """
              for value in range(self.starts[index], self.AP_Peak_Times[index]):
                if np.allclose(self.data[value], half_AP_height, atol=1.5) is True:
                  cursors.append(value)
                  print "Success"
                  break

              for value in range(self.AP_Peak_Times[index], self.ends[index]):
                if np.allclose(self.data[value], half_AP_height, atol=1.5) is True:
                  cursors.append(value)
                  print "Success"
                  break

            """



        else:
          self.AHP_Durations = [0,0,0]
          self.AHP_Peak_Times = [0,0,0]
          self.AHP_Peak_Amplitudes = [0,0,0]
          self.AHP_Max_Rises = [0,0,0]
          self.AHP_Max_Decays = [0,0,0]
          self.AHP_Fast_Troughs = [0,0,0]
          self.AHP_Slow_Troughs = [0,0,0]
          self.AHP_Slow_Trough_Times = [0,0,0]
          self.AHP_Absolute_Peaks = [0,0,0]
          self.AP_Heights = [0,0,0]
          self.AP_Half_Height_Widths = [0,0,0]
```

```
def washon_washoff(self, thresh, start):
    #create new df using self.index, create new columns creating list from number of traces in file
    #loop through sweeps, for n in range get filesize
    #perform AP analysis on all sweeps
    #add x to df
    #select from data frame key measures, firing rate, amplitude etc and plt over time
    size=stf.get_size_channel()
    columns=list(range(size))

    cont_df=pd.DataFrame(index=self.index, columns=columns)
    for n in range(1, size):
        self.AP_diff(n)
        self.baseline()
        self.AP_start(thresh, start)
        self.AP_end()
        self.AHP_end()
        self.AP_analysis()
        self.AHP()


        x = [self.thresholds[0], self.Potentials, self.AP_Absolute_Peaks[0], self.AP_Peak_Times[0],
             self.AP_Peak_Amplitudes[0],
             self.AP_Durations[0], self.AP_Depol_Times[0], self.AP_Repol_Times[0], self.AP_Max_Rises[0],
             self.AP_Max_Decays[0],
             self.AP_Rise_Decay_Ratio[0], self.first_AP_data, self.first_AP_array,
             self.AP_interspike_intervals[0],
             self.Av_ISI, self.First_ISI, self.ISI_Coeff_Vars, self.Num_Of_AP, self.Av_Firing_Freq,
             self.Check_Av_Firings, self.Instantaneous_Firing_Freq, self.Latencies,
             self.Adaptation_Indexes, self.Delay, self.Burst, self.Pause, self.Steady_Firing_Freq,
             self.AHP_Durations[0], self.AHP_Peak_Times[0], self.AHP_Peak_Amplitudes[0],
             self.AHP_Absolute_Peaks[0],
             self.AHP_Max_Rises[0], self.AHP_Max_Decays[0], self.AHP_Fast_Troughs[0],
             self.AHP_Slow_Troughs[0],
             self.AHP_Slow_Trough_Times[0], self.AP_Heights[0], self.AP_Half_Height_Widths[0],
             self.Peak_Deflection,
             self.Peak_Deflection_Amplitude, self.Peak_Deflection_Time, self.Steady_Deflection,
             self.Steady_Deflection_Amplitude, self.Sag, self.ten_percent_max,
             self.ten_percent_max_time, self.membrane_constant]

        cont_df[str(n)]= x

    AP_amplitude=cont_df.loc["AP_Peak_Amplitudes"]
    AP_duration=cont_df.loc["AP_Durations"]
    Firing_rate=cont_df.loc["Check_Av_Firings"]
    threshold=cont_df.loc["thresholds"]
    depol_time=cont_df.loc["AP_Depol_Times"]
    repol_time=cont_df.loc["AP_Repol_Times"]
    max_rise=cont_df.loc["AP_Max_Rises"]
    max_decay=cont_df.loc["AP_Max_Decays"]
    AHP_duration=cont_df.loc["AHP_Durations"]
    AHP_amplitude=cont_df.loc["AHP_Peak_Amplitudes"]

    b=[AP_amplitude, AP_duration, Firing_rate, threshold, depol_time, repol_time, max_rise, max_decay,
       AHP_duration, AHP_amplitude]
    for a in b:
        plt.figure()
        x=list(range(len(a)))
        y=a
        plt.plot(x, y)
        plt.show()
```

Appendix III- Plasmids

Kv3.1b

Accession number

NM_004976.2

Plasmid sequence

aacaaaatattaacgcttacaatttccattcgccattcaggctgcgcaactgttgggaagggcgatcggtgcgggcctcttcgctattacgccagctggcgaaaggggatgtgctgcaaggcgattaagttgggtaacgccagggttttccca
gtcacgacgttgtaaaacgacggccagtgccaagctgatctatacattgaatcaatattggcaattagccatattagtcattggttatatagcataaatcaatattggctattggccattgcatacgttgtatctatatcataatatgtacatttata
ttggctcatgtccaatatgaccgccatgttgacattgattattgactagttattaatagtaatcaattacggggtcattagttcatagcccatatatggagttccgcgttacataacttacggtaaatggcccgcctggctgaccgcccaacgaccc
ccgcccattgacgtcaataatgacgtatgttcccatagtaacgccaatagggactttccattgacgtcaatgggtggagtatttacggtaaactgcccacttggcagtacatcaagtgtatcatatgccaagtccgcccccctattgacgtcaatg
acggtaaatggcccgcctggcattatgcccagtacatgaccttacgggactttcctacttggcagtacatctacgtattagtcatcgctattaccatggtgatgcggttttggcagtacaccaatgggcgtggatagcggtttgactcacggggat
ttccaagtctccaccccattgacgtcaatgggagtttgttttggcaccaaaatcaacgggactttccaaaatgtcgtaataaccccgccccgttgacgcaaatgggcggtaggcgtgtacggtgggaggtctatataagcagagctcgtttagt
gaaccgtcagaattttgtaatacgactcactatagggcggccgggaattcgtcgactggatccggtaccgaggagatctgccgccgcgatcgccatgggccaaggggacgagagcgagcgcatcgtgatcaacgtgggcggcacgcgcca
ccagacgtaccgctcgaccctgcgcacgctgcccggcacgcggctcgcctggctggcggagcccgacgcccacagccacttcgactatgacccgcgtgctgacgagttcttcttcgaccgccaccccggcgtcttcgcgcacatcctgaacta
ctaccgcacgggcaagctgcactgcccagccgacgtgtgcgggccgctctacgaggaggagctggccttctggggcatcgacgagaccgacgtggagccctgctgctggatgacgtaccgccagcaccgcgacgccgaggaggctctgga
cagcttcggcggcgctcctctggacaacagcgccgacgacgcggacgccgacggccctggcgactcgggcgacggcgaggacgagctggagatgaccaagcgcctggcgctcagtgactccccggatggccggcctggcggcttttggcg
ccgctggcagccgcgcatctgggcgctcttcgaggacccgtactcgtcccgctacgcgcggtatgtggccttcgcttccctcttcttcatcctggtctccatcaccaccttctgcctggagacccacgagcgcttcaaccccatcgtgaacaagac

ggagatcgagaacgttcgcaatggcacgcaagtgcgctactaccgggaggccgagacggaggccttccttacctacatcgagggcgtctgtgtggtctggttcaccttcgagttcctcatgcgtgtcatcttctgccccaacaaggtagagttc
atcaagaactcgctcaacatcattgactttgtggccatcctgcccttctacctggaggtggggctgagcggcctgtcctccaaggcagccaaggacgtgctgggcttcctgcgcgtcgtccgcttcgtgcgcatcttgcgcatctttaagctgacc
cgccactttgtgggcctgcgggtcctgggccacacgctccgagccagcaccaacgagttcctgctgctcatcatcttcctggccttgggcgtgctgatcttcgccaccatgatctactacgccgagaggatagggcacagcccaatgacccca
gcgccagtgagcacacgcactttaagaacatccccatcggcttctggtgggccgtggtcaccatgacgaccctgggctatggagacatgtacccgcagacgtggtccggcatgctggtgggggctctgtgtgcgctggcgggcgtgctcacc
atcgccatgcccgtgcccgtcatcgtgaacaatttcgggatgtattactccttagccatggctaagcagaaactaccaaagaaaaaaaagaagcatattccgcggccaccgcagctgggatctcccaattattgtaaatctgtcgtaaactctc
cacaccacagtactcagagtgacacatgtccgctggcccaggaagaaattttagaaattaacagagcaggtaggaaacctcttagaggcatgtcgatcagcggaccgacgcgtacgcggccgctcgagcagaaactcatctcagaagagg
atctggcagcaaatgatatcctggattacaaggatgacgacgataaggtttaaacggccggccgcggtcatagctgtttcctgaacagatcccgggtggcatccctgtgacccctccccagtgcctctcctggccctggaagttgccactccag
tgcccaccagccttgtcctaataaaattaagttgcatcattttgtctgactaggtgtccttctataatattatggggtggaggggggtggtatggagcaaggggcaagttgggaagacaacctgtagggcctgcggggtctattgggaaccaag
ctggagtgcagtggcacaatcttggctcactgcaatctccgcctcctgggttcaagcgattctcctgcctcagcctcccgagttgttgggattccaggcatgcatgaccaggctcagctaatttttgtttttttggtagagacggggtttcaccatat
tggccaggctggtctccaactcctaatctcaggtgatctacccaccttggcctcccaaattgctgggattacaggcgtgaaccactgctcccttccctgtccttctgattttaaaataactataccagcaggaggacgtccagacacagcatagg
ctacctggccatgcccaaccggtgggacatttgagttgcttgcttggcactgtcctctcatgcgttgggtccactcagtagatgcctgttgaattgggtacgcggccagcggcgagcggtatcagctcactcaaaggcggtaatacggttatcca
cagaatcaggggataacgcaggaaagaacatgtgagcaaaaggccagcaaaaggccaggaaccgtaaaaaggccgcgttgctggcgtttttccataggctccgcccccctgacgagcatcacaaaaatcgacgctcaagtcagaggtgg
cgaaacccgacaggactataaagataccaggcgtttccccctggaagctccctcgtgcgctctcctgttccgaccctgccgcttaccggatacctgtccgcctttctcccttcgggaagcgtggcgctttctcatagctcacgctgtaggtatctc
agttcggtgtaggtcgttcgctccaagctgggctgtgtgcacgaaccccccgttcagcccgaccgctgcgccttatccggtaactatcgtcttgagtccaacccggtaagacacgacttatcgccactggcagcagccactggtaacaggatta
gcagagcgaggtatgtaggcggtgctacagagttcttgaagtggtggcctaactacggctacactagaagaacagtatttggtatctgcgctctgctgaagccagttaccttcggaaaaagagttggtagctcttgatccggcaaacaaacca
ccgctggtagcggtggtttttttgtttgcaagcagcagattacgcgcagaaaaaaaggatctcaagaagatcctttgatcttttctacggggtctgacgctcagtggaacgaaaactcacgttaagggattttggtcatgagattatcaaaaagg
atcttcacctagatccttttaaattaaaaatgaagttttaaatcaatctaaagtatatatgagtaacctgaggctatggcagggcctgccgccccgacgttggctgcgagccctgggccttcacccgaacttgggggtggggtggggaaaagg
aagaaacgcgggcgtattggccccaatggggtctcggtggggtatcgacagagtgccagccctgggaccgaaccccgcgtttatgaacaaacgacccaacaccgtgcgttttattctgtctttttattgccgtcatagcgcgggttccttccggt
attgtctccttccgtgtttcagttagcctccccctagggtgggcgaagaactccagcatgagatccccgcgctggaggatcatccagccggcgtcccggaaaacgattccgaagcccaacctttcatagaaggcggcggtggaatcgaaatct
cgtgatggcaggttgggcgtcgcttggtcggtcatttcgaaccccagagtcccgctcagaagaactcgtcaagaaggcgatagaaggcgatgcgctgcgaatcgggagcggcgataccgtaaagcacgaggaagcggtcagcccattcgc
cgccaagctcttcagcaatatcacgggtagccaacgctatgtcctgatagcgatccgccacacccagccggccacagtcgatgaatccagaaaagcggccattttccaccatgatattcggcaagcaggcatcgccatgggtcacgacgaga
tcctcgccgtcgggcatgctcgccttgagcctggcgaacagttcggctggcgcgagcccctgatgctcttcgtccagatcatcctgatcgacaagaccggcttccatccgagtacgtgctcgctcgatgcgatgtttcgcttggtggtcgaatgg
gcaggtagccggatcaagcgtatgcagccgccgcattgcatcagccatgatggatactttctcggcaggagcaaggtgagatgacaggagatcctgccccggcacttcgcccaatagcagccagtcccttcccgcttcagtgacaacgtcga
gcacagctgcgcaaggaacgcccgtcgtggccagccacgatagccgcgctgcctcgtcttgcagttcattcagggcaccggacaggtcggtcttgacaaaaagaaccgggcgcccctgcgctgacagccggaacacggcggcatcagagc
agccgattgtctgttgtgcccagtcatagccgaatagcctctccacccaagcggccggagaacctgcgtgcaatccatcttgttcaatcatgcgaaacgatcctcatcctgtctcttgatcgatctttgcaaaagcctaggcctccaaaaaagcc
tcctcactacttctggaatagctcagaggccgaggcggcctcggcctctgcataaataaaaaaaattagtcagccatggggcggagaatgggcggaactgggcggagttaggggcgggatgggcggagttaggggcgggactatggttgct
gactaattgagatgcatgctttgcatacttctgcctgctggggagcctggggactttccacacctggttgctgactaattgagatgcatgctttgcatacttctgcctgctggggagcctggggactttccacaccctaactgacacacattccaca
gctggttctttccgcctcaggactcttcctttttcaatattattgaagcatttatcagggttattgtctcatgagcggatacatatttgaatgtatttagaaaaataaacaaataggggttccgcgcacatttccccgaaaagtgccacctgacgcgc
cctgtagcggcgcattaagcgcggcgggtgtggtggttacgcgcagcgtgaccgctacacttgccagcgccctagcgcccgctcctttcgctttcttcccttcctttctcgccacgttcgccggctttccccgtcaagctctaaatcgggggctccc
tttagggttccgatttagtgctttacggcacctcgaccccaaaaaacttgattagggtgatggttcacgtagtgggccatcgccctgatagacggtttttcgccctttgacgttggagtccacgttctttaatagtggactcttgttccaaactggaa
caacactcaaccctatctcggtctattcttttgatttataagggattttgccgatttcggcctattggttaaaaaatgagctgatttaacaaaaatttaacgcgaatttt

Protein sequence

MGQGDESERIVINVGGTRHQTYRSTLRTLPGTRLAWLAEPDAHSHFDYDPRADEFFFDRHPGVFAHILNYYRTGKLHCPADVCGPLYEEELAFWGIDETDVEPCCWMTYRQHR
DAEEALDSFGGAPLDNSADDADADGPGDSGDGEDELEMTKRLALSDSPDGRPGGFWRRWQPRIWALFEDPYSSRYARYVAFASLFFILVSITTFCLETHERFNPIVNKTEIENVR
NGTQVRYYREAETEAFLTYIEGVCVVWFTFEFLMRVIFCPNKVEFIKNSLNIIDFVAILPFYLEVGLSGLSSKAAKDVLGFLRVVRFVRILRIFKLTRHFVGLRVLGHTLRASTNEFLL
LIIFLALGVLIFATMIYYAERIGAQPNDPSASEHTHFKNIPIGFWWAVVTMTTLGYGDMYPQTWSGMLVGALCALAGVLTIAMPVPVIVNNFGMYYSLAMAKQKLPKKKKKHIPR
PPQLGSPNYCKSVVNSPHHSTQSDTCPLAQEEILEINRAGRKPLRGMSI

Kv3.4a

Accession number

NM_004978.4

Plasmid sequence

cacgacggggagtcaggcaactatggatgaacgaaatagacagatcgctgagataggtgcctcactgattaagcattggtaactgtcagaccaagtttactcatatatactttagattgatttaaaacttcatttttaatttaaaaggatctaggt
gaagatcctttttgataatctcatgaccaaaatcccttaacgtgagttttcgttccactgagcgtcagaccccgtagaaaagatcaaaggatcttcttgagatcctttttttctgcgcgtaatctgctgcttgcaaacaaaaaaaccaccgctacca
gcggtggtttgtttgccggatcaagagctaccaactctttttccgaaggtaactggcttcagcagagcgcagataccaaatactgtccttctagtgtagccgtagttaggccaccacttcaagaactctgtagcaccgcctacatacctcgctctg
ctaatcctgttaccagtggctgctgccagtggcgataagtcgtgtcttaccgggttggactcaagacgatagttaccggataaggcgcagcggtcgggctgaacggggggttcgtgcacacagcccagcttggagcgaacgacctacaccga
actgagatacctacagcgtgagcattgagaaagcgccacgcttcccgaagggagaaaggcggacaggtatccggtaagcggcagggtcggaacaggagagcgcacgagggagcttccagggggaaacgcctggtatctttatagtcctg
tcgggtttcgccacctctgacttgagcgtcgatttttgtgatgctcgtcaggggggcggagcctatggaaaaacgccagcaacgcggcctttttacggttcctggccttttgctggccttttgctcacatgttctttcctgcgttatcccctgattctg
tggataaccgtattaccgcctttgagtgagctgataccgctcgccgcagccgaacgaccgagcgcagcgagtcagtgagcgaggaagcggaagagcgcccaatacgcaaaccgcctctccccgcgcgttggccgattcattaatgcagagc
ttgcaattcgcgcgtttttcaatattattgaagcatttatcagggttattgtctcatgagcggatacatatttgaatgtatttagaaaaataaacaaataggggttccgcgcacatttccccgaaaagtgccacctgacgtctaagaaaccattatt
atcatgacattaacctataaaaataggcgtagtacgaggccctttcactcattagatgcatgtcgttacataacttacggtaaatggcccgcctggctgaccgcccaacgacccccgcccattgacgtcaataatgacgtatgttcccatagtaa
cgccaatagggactttccattgacgtcaatgggtggagtatttacggtaaactgcccacttggcagtacatcaagtgtatcatatgccaagtacgccccctattgacgtcaatgacggtaaatggcccgcctggcattatgcccagtacatgacc
ttatgggactttcctacttggcagtacatctacgtattagtcatcgctattaccatggtgatgcggttttggcagtacatcaatgggcgtggatagcggtttgactcacggggatttccaagtctccaccccattgacgtcaatgggagtttgttttg
gcaccaaaatcaacgggactttccaaaatgtcgtaacaactccgccccattgacgcaaatgggcggtaggcgtgtacggtgggaggtctatataagcagagctctccctatcagtgatagagatctccctatcagtgatagagatcgtcgacg
agctcgtttagtgaaccgtcagatcgcctggagacgccatccacgctgttttgacctccatagaagacaccgggaccgatccagcctccggactctagaggatccctaccggtgatatcctcgagcccatcaacaagtttgtacaaaaaagca
ggcttcaccatgatcagctcggtgtgtgtctcctcctaccgcgggcgcaagtcggggaacaagcctccgtccaaaacatgtctgaaggaggagatggccaagggcgaggcgtcggagaagatcatcatcaacgtgggcggcacgcgacat
gagacctaccgcagcaccctgcgcaccctaccgggaacccgcctcgcctggctggccgaccccgacggcgggggccggcccgagaccgatggcggcggtgtgggtagcagcggcagcagcggcggcgggggctgcgagttcttcttcga
caggcacccgggcgtcttcgcctacgtgctcaactactaccgcaccggcaagctgcactgccccgcggacgtgtgcgggccgctcttcgaagaggagctcaccttctggggcatcgacgagaccgacgtggaaccctgctgctggatgacct
accggcagcaccgcgacgccgaggaggcgctcgacatcttcgagagcccggacggaggcggcagcggcgcggggcccagcgacgaggccggcgacgatgagcgggagctggccctgcagcgactgggcccccacgagggaggcgcg
ggccatggcgccgggtctgggggctgccgcggctggcagccccgcatgtgggcgctcttcgaggatccctactcctcccgggccgctagggtagtggcctttgcctctctcttcttcatcctggtctccatcaccactttctgcctggagacccat
gaggcctttaatatcgaccgcaacgtgacagagatcctccgcgtagggaacatcaccagcgtgcacttccggcgggaggtagagacagagcccatcctgacctacatcgagggcgtatgtgtgctgtggttcacactggagttcctggtgcg
catcgtgtgctgccccgacacgctggacttcgtcaagaacctgctcaacatcatcgactttgtggccatcctgcccttctacctggaggtggggctgagcggcctgtcatccaaggcggcccgcgacgtgctgggcttcctgcgcgtggtgcgct
tcgtgcgcatcctgcgtatcttcaagctcacacgccacttcgtggggctacgcgtgctgggccacaccctgagggccagcaccaatgagttcctgctgcttatcatcttcctggccctgggtgtgctcatctttgccaccatgatctactacgctga
gcgcattggggccaggccctccgaccctcggggtaatgaccacaccgacttcaagaacatccccattggcttctggtgggctgtggtcaccatgacgacactgggctacggagacatgtaccccaagacgtggtcaggcatgctggtagggg
cactgtgtgcactggctggcgtgctcaccatcgccatgccggtgcctgtcatcgtcaacaacttcggcatgtactactccctggccatggccaagcagaagctgcccaagaaacggaagaagcacgtgccacggccggcgcagctggagtca
cccatgtactgcaagtctgaggagacttcccccgggacagcacctgcagtgataccagccccctgcccgggaagagggtatgatcgagaggaaacgggcagactctaagcagaatggcgatgccaacgcagtgctgtctgatgaggag
ggagctggcctcacccaacccctggcctcctccccgacccccgaggagcgccgggccctgcgacgctccaccactcgagacagaaacaagaaggcagctgcctgcttcctgctcagcactggggactatgcctgcgccgatggtagtgtcc
ggaaaggcacattcgtcctccgtgaccttccccttcagcattcacctgaggctgcatgccctccaactgctgggactctgttcctgccacattgagacccagctttcttgtacaaagtggttgatgggcggccgctctagagggcccaagcttac

gcgtgcatgcgacgtcatagctctctccctatagtgagtcgtattataagctaggcactggccgtcgttttacaacgtcgtgactgggaaaactgctagcttgggatctttgtgaaggaaccttacttctgtggtgtgacataattggacaaacta
cctacagagatttaaagctctaaggtaaatataaaatttttaagtgtataatgtgttaaactagctgcatatgcttgctgcttgagagttttgcttactgagtatgatttatgaaaatattatacacaggagctagtgattctaattgtttgtgtatttt
agattcacagtcccaaggctcatttcaggcccctcagtcctcacagtctgttcatgatcataatcagccataccacatttgtagaggtttttacttgctttaaaaaacctcccacacctcccctgaacctgaaacataaaatgaatgcaattgttgt
tgttaacttgtttattgcagcttataatggttacaaataaagcaatagcatcacaaatttcacaaataaagcatttttttcactgcattctagttgtggtttgtccaaactcatcaatgtatcttatcatgtctggatcgatcctgcattaatgaatcgg
ccaacgcgcggggagaggcggtttgcgtattggctggcgtaatagcgaagaggcccgcaccgatcgcccttcccaacagttgcgcagcctgaatggcgaatgggacgcgccctgtagcggcgcattaagcgcggcgggtgtggtggttacg
cgcagcgtgaccgctacacttgccagcgccctagcgcccgctcctttcgctttcttcccttcctttctcgccacgttcgccggctttccccgtcaagctctaaatcgggggctcccttagggttccgatttagtgctttacggcacctcgaccccaa
aaaacttgattagggtgatggttcacgtagtgggccatcgccctgatagacggtttttcgccctttgacgttggagtccacgttctttaatagtggactcttgttccaaactggaacaacactcaaccctatctcggtctattcttttgatttataagg
gattttgccgatttcggcctattggttaaaaaatgagctgatttaacaaatatttaacgcgaattttaacaaaatattaacgtttacaatttcgcctgatgcggtattttctccttacgcatctgtgcggtatttcacaccgcatacgcggatctgcgc
agcaccatggcctgaaataacctctgaaagaggaacttggttaggtaccttctgaggcggaaagaaccagctgtggaatgtgtgtcagttagggtgtggaaagtccccaggctccccagcaggcagaagtatgcaaagcatgcatctcaatt
agtcagcaaccaggtgtggaaagtccccaggctccccagcaggcagaagtatgcaaagcatgcatctcaattagtcagcaaccatagtcccgccccttaactccgcccatcccgcccctaactccgcccagttccgcccattctccgccccatg
gctgactaatttttttttatttatgcagaggccgaggccgcctcggcctctgagctattccagaagtagtgaggaggctttttggaggcctaggcttttgcaaaaagcttgattcttctgacacaacagtctcgaacttaaggctagagccaccat
gattgaacaagatggattgcacgcaggttctccggccgcttgggtggagaggctattcggctatgactgggcacaacagacaatcggctgctctgatgccgccgtgttccggctgtcagcgcaggggcgcccggttcttttgtcaagaccgac
ctgtccggtgccctgaatgaactgcaggacgaggcagcgcggctatcgtggctggccacgacgggcgttccttgcgcagctgtgctcgacgttgtcactgaagcgggaagggactggctgctattgggcgaagtgccggggcaggatctcct
gtcatctcaccttgctcctgccgagaaagtatccatcatggctgatgcaatgcggcggctgcatacgcttgatccggctacctgcccattcgaccaccaagcgaaacatcgcatcgagcgagcacgtactcggatggaagccggtcttgtcgat
caggatgatctggacgaagagcatcaggggctcgcgccagccgaactgttcgccaggctcaaggcgcgcatgcccgacggcgaggatctcgtcgtgacccatggcgatgcctgcttgccgaatatcatggtggaaaatggccgcttttctgg
attcatcgactgtggccggctgggtgtggcggaccgctatcaggacatagcgttggctacccgtgatattgctgaagagcttggcggcgaatgggctgaccgcttcctcgtgctttacggtatcgccgctcccgattcgcagcgcatcgccttct
atcgccttcttgacgagttcttctgagcgggactctggggttcgaaatgaccgaccaagcgacgcccaacctgccatcacgatggccgcaataaaatatctttattttcattacatctgtgttgttggtttttgtgtgaatcgatagcgataaggat
ccgcgtatggtgcactctcagtacaatctgctctgatgccgcatagttaagccagcccgacacccgccaacacccgctgacgcgccctgacgggcttgtctgctcccggcatccgcttacagacaagctgtgaccgtctccgggagctgcatg
tgtcagaggttttcaccgtcatcaccgaaacgcgcgagacgaaagggcctcgtgatacgcctatttttataggttaatgtcatgataataatggtttcttagacgtcaggtggcacttttcggggaaatgtgcgcggaacccctatttgtttatttttt
ctaaatacattcaaatatgtatccgctcatgagacaataaccctgataaatgcttcaataatattgaaaaaggaagagtatgagtattcaacatttccgtgtcgcccttattcccttttttgcggcattttgccttcctgtttttgctcacccagaaac
gctggtgaaagtaaaagatgctgaagatcagttgggtgcacgagtgggttacatcgaactggatctcaacagcggtaagatccttgagagttttcgcccgaagaacgttttccaatgatgagcacttttaaagttctgctatgtggcgcggta
ttatcccgtattgacgccgggcaagagcaactcggtcgccgcatacactattctcagaatgacttggttgagtactcaccagtcacagaaaagcatcttacggatggcatgacagtaagagaattatgcagtgctgccataaccatgagtgata
acactgcggccaacttacttctgacaacgatcggaggaccgaaggagctaaccgcttttttgcacaacatgggggatcatgtaactcgccttgatcgttgggaaccggagctgaatgaagccataccaaacgacgagcgtgacaccacgatg
cctgtagcaatggcaacaacgttgcgcaaactattaactggcgaactacttactctagcttcccggcaacaattaatagactggatggaggcggataaagttgcaggaccacttctgcgctcggcccttccggctggctggtttattgctgataa
atctggagccggtgagcgtgggtctcgcggtatcattgcagcactggggccagatggtaagccctcccgtatcgtagttatcta

Protein sequence

MISSVCVSSYRGRKSGNKPPSKTCLKEEMAKGEASEKIIINVGGTRHETYRSTLRTLPGTRLAWLADPDGGGRPETDGGGVGSSGSSGGGGCEFFFDRHPGVFAYVLNYYRTGKLH
CPADVCGPLFEEELTFWGIDETDVEPCCWMTYRQHRDAEEALDIFESPDGGGSGAGPSDEAGDDERELALQRLGPHEGGAGHGAGSGGCRGWQPRMWALFEDPYSSRAARVV
AFASLFFILVSITTFCLETHEAFNIDRNVTEILRVGNITSVHFRREVETEPILTYIEGVCVLWFTLEFLVRIVCCPDTLDFVKNLLNIIDFVAILPFYLEVGLSGLSSKAARDVLGFLRVV
RFVRILRIFKLTRHFVGLRVLGHTLRASTNEFLLLIIFLALGVLIFATMIYYAERIGARPSDPRGNDHTDFKNIPIGFWWAVVTMTTLGYGDMYPKTWSGMLVGALCALAGVLTIA
MPVPVIVNNFGMYYSLAMAKQKLPKKRKKHVPRPAQLESPMYCKSEETSPRDSTCSDTSPPAREEGMIERKRADSKQNGDANAVLSDEEGAGLTQPLASSPTPEERRALRRSTT
RDRNKKAAACFLLSTGDYACADGSVRKGTFVLRDLPLQHSPEAACPPTAGTLFLPH*