Mapping, tracking and modeling the movements of single membrane-bound transcription activator proteins in live *Vibrio cholerae*

by

Beth L. Haas

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Chemistry)
in the University of Michigan
2014

Doctoral Committee:

Assistant Professor Julie S. Biteen, Chair
Professor Victor J. DiRita
Assistant Professor Sarah L. Veatch
Professor Nils G. Walter

"The laws of nature are not discovered by accident; theories do not come by chance, even to the greatest minds; they are not born in the hurry and worry of daily toil; they are diligently sought; they are patiently waited for, they are received with cautious reserve, they are accepted with reverence and awe."

— Maria Mitchell,
Sweeper in the Sky by Helen Wright

In memory of my grandfathers,
Dr. Seeley M. Phillips
and
Dr. John E. Leverett,
who valued curiosity,
story-telling,
and hard work.

Acknowledgments

I owe thanks to so many people, starting with my advisor, Julie Biteen, who has built a pretty awesome lab that I feel very lucky to be a part of.

Many thanks to my awesome labmates: Mou-Chi, who always has the best advice and to whom I owe so many lunches for his wisdom and experience; Esther, who reminds me to stay balanced; Jess, whose cheerfulness is infectious; Chanrith, who has been a fun and funny officemate, and whom I trust will do great things with the *Vibrio* project; Yi and Dave, who have shared many a helpful MATLAB trick; and Hannah, Bing, Sherry, Ben, and the rest of our rotating crew, for good times in lab and out.

Thanks also to former lab members Krish Karunatilaka, for sound advice and her ability to find a silver lining in every cloud; John Jurkas, who brought a nearly overwhelming amount of enthusiasm to our work together; Aaron Konopko, for assisting with experiments and letting me bombard him with questions over lunch; and Ben Coupland, who remembers The Beginning, the boxes, and just how many times we reassembled the laser table the first year.

Thanks to Sarah Veatch and Nils Walter, for good advice and guidance as members of my committee. Special thanks to Vic DiRita, my collaborator and committee member, who teaches me something new every time we meet.

Thanks to my collaborator Jyl Matson, who makes genetics seem like magic.

Thanks to the bioimaging journal club crew, especially Elín Edwald and Matt Stone, who have shared some fascinating work and with whom it has been a pleasure to discuss new ideas.

Thanks to my Ann Arbor friends, particularly the GradTONES, who have helped me relax and recharge, and my "lunch ladies" Rachel Barnard and Sabrina Peczoncyzk, who have been sharing, commiserating and celebrating with me since Day One.

Thanks to my family, who cheer me on, even when they have no idea what it is I do.

Most of all, thanks to my husband, Kevin, for mugs of tea and loaves of bread, for staying up late and getting up early, for his patience, and laughter, and so much love.

Table of Contents

List of Tables

List of Figures

Abstract

The human disease cholera has been known since ancient times, yet there is still no cure. *Vibrio cholerae* bacteria cause cholera by producing the deadly cholera toxin. Cholera toxin production is regulated by the protein ToxT, which itself is regulated by two membrane-bound proteins, TcpP and ToxR, that work together to bind DNA and activate transcription of *toxT*. The molecular-scale details of this unusual membrane-bound transcription activation mechanism are unclear and cannot be observed using traditional light microscopy due to the diffraction limit of light.

In this thesis, we use single-molecule tracking and super-resolution localization microscopy to overcome the diffraction limit and directly observe the motions and interactions of TcpP labeled with the fluorescent proteins Dendra2, mCherry and PAmCherry in live *V. cholerae* cells. We describe methods developed–and obstacles encountered–in the course of our studies of these protein fusions, and we find that, of the three fluorescent proteins examined, PAmCherry is the best choice for tracking TcpP motion.

By using mean squared displacement and cumulative probability distribution analyses of single-molecule trajectories, we compare TcpP-PAmCherry motions across three *V. cholerae* strains. In each strain, the native copy of TcpP has been removed and replaced with TcpP-PAmCherry expressed ectopically. We find that TcpP can be categorized into three populations: fast, slow and immobile; and that TcpP-PAmCherry moves faster when both binding partners (ToxR and the *toxT* promoter) are present than when either is lacking ($\Delta toxR$ or $toxT_{\Delta\text{pro}}$). Our findings support a mechanism for TcpP–ToxR–*toxT* promoter interaction in which ToxR recruits TcpP to the *toxT* promoter.

Although PAmCherry is adequate for our single-molecule microscopy experiments, it is not an ideal fluorophore. A brighter fluorescent label that resists photobleaching would enable faster imaging and longer measured trajectories. We present a protocol for enhancing PAmCherry fluorescence by coupling TcpP-PAmCherry in the membrane of live *V. cholerae* cells to extracellular gold nanoisland films and nanotriangle arrays to achieve plasmon-enhanced fluorescence. We find that single-molecule fluorescence can be detected above background scatter on both nanostructured gold surfaces, which is promising for further live cell studies.

Greater understanding of the ToxR regulon may lead to novel therapeutics to combat cholera, and enhanced fluorescence will help us observe such interactions with greater detail.

Chapter 1

Introduction

We live in a world dominated by microbes. It is estimated that bacterial cells in the body outnumber our own human cells by ten to one (Savage, 1977). Bacteria make us sick (Kaper et al., 1994), but they also keep us well (Koropatkin et al., 2008). Despite their abundance and importance, there is much we still do not understand about them. Bacteria are very small (typically 1–10 µm long), and the molecules inside them are even smaller (0.5–10 nm). Direct observation of molecular dynamics is difficult to achieve at such a scale, but new methods that can reach molecular-scale resolution promise to make the hidden world inside bacteria accessible to us. This thesis describes the application of super-resolution imaging to the specific case of understanding the motions of the protein TcpP in live *Vibrio cholerae* bacteria, and how we can reach imaging resolutions approaching the scale of the molecules themselves.

1.1 The diffraction limit

Light microscopy is non-invasive, minimally perturbative, and compatible with live samples. However, nearly 150 years ago, Ernst Abbe discovered how the diffraction of light limits the resolution of a microscope (Abbe, 1873). No matter how powerful the magnification of the objective lens, even an infinitesimally small point source will produce an image of finite size, called the point-spread function (PSF). Two points placed very close together will have overlapping PSFs. The minimum separation distance at which the points can be resolved is the Abbe limit (Λ_{min}), which depends on the wavelength (λ) of the imaging light (390–700 nm for light in the visible spectrum) and the numerical aperture (NA) of the objective used, according to the following equation:

$$\Lambda_{\text{min}} = \frac{\lambda}{2n \sin \alpha} = \frac{\lambda}{2\text{NA}} \tag{1.1}$$

where n is the index of refraction of the imaging medium and α is the half-cone angle of the observable area for the objective (Fig. 1.1). For example, using a yellow-green laser

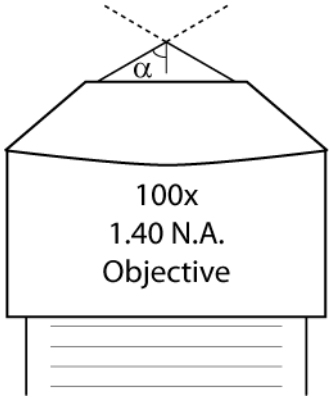

Figure 1.1: The half-cone angle, α, for an objective

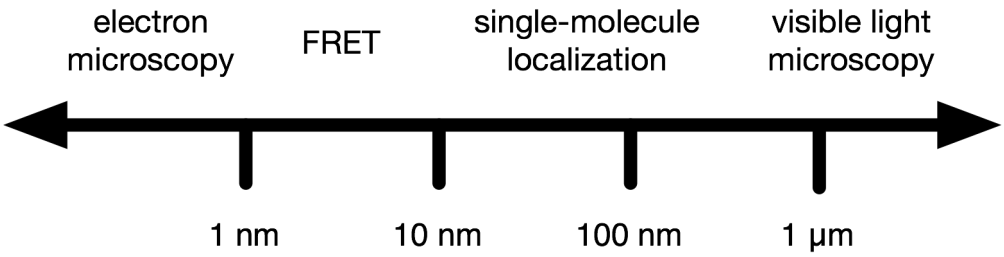

Figure 1.2: Resolvable size scales using several microscopy techniques

($\lambda = 561$ nm) and an objective with a large NA (e.g. 1.4), the smallest resolvable object is approximately 200 nm in size.

Greater precision can be achieved by using smaller wavelengths; for example, an electron microscope can reach sub-nanometer resolution (Fig. 1.2; Chiu et al., 2005), but such methods are not compatible with live biological samples and dynamic information is lost. Other methods, such as Förster Resonance Energy Transfer (FRET), can report on the nanoscale proximity of two fluorescent labels, but to date FRET has found limited applications in live bacteria. This thesis presents an alternative approach to overcoming the diffraction limit of visible light by using super-resolution localization microscopy and single-molecule tracking methods (Qian et al., 1991; Betzig, 1995; Schütz et al., 1997; Moerner and Orrit, 1999; van Oijen et al., 1999; Kubitscheck et al., 2000). Such approaches achieve spatial and temporal resolutions of tens of nanometers and tens of milliseconds in live cells (Kim et al., 2006; Yu et al., 2006; van den Wildenberg et al., 2011; Bakshi et al., 2012). These methods enable the direct observation of molecular movements and interactions too small to see using traditional light microscopy, and they offer additional ways to answer biological and biophysical questions through direct observation. Already super-resolution microscopy has been used to probe the structural protein MreB in live *Caulobacter crescentus* (Kim et al., 2006; Biteen et al., 2008), the diffusion of the *Escherichia coli* membrane pore protein TatA (van den Wildenberg et al., 2011), and the dynamics of the *lac* repressor in live *E. coli* (Elf et al., 2007). Beyond bacteria, these

methods have found applications as varied as mitochondrial membranes in HeLa cells (Appelhans et al., 2012), molecular motors *in vitro* (Yildiz et al., 2003), and the actin cytoskeleton in kidney cells (Xu et al., 2012).

1.2 Super-resolution fluorescence microscopy

Individual molecules are more than an order of magnitude too small to observe with traditional microscopy methods. Our goal, then, is to use super-resolution imaging to determine molecular positions with precisions on the same 1–10-nm scale as the biomolecules themselves. By labeling molecules of interest with fluorescent tags (Fernández-Suárez and Ting, 2008), single molecules can be detected and localized above a non-fluorescent or dimly fluorescent background if they are well separated from each other. A fluorescent molecule can be approximated as a point emitter, with a position corresponding to the center of its intensity distribution. This intensity distribution for each fluorophore is fit with a model function. The actual intensity distribution is an Airy disk (Airy, 1835), but a two-dimensional Gaussian function is commonly used as an approximation of the central lobe of an Airy disk due to its similarity in shape and relative simplicity in calculation (Fig. 1.3; Thompson et al., 2002). Fluorescence is collected by a pixellated detector, such as an electron multiplying charge-coupled device (EMCCD) camera.

The precision with which a molecule can be localized (Δx) is dependent on several factors, including the size of the pixels in the detector (a), the background noise per pixel (b), the standard deviation of the model point-spread function (s), which is related to the Abbe limit (Eq. 1.1), and the number of photons detected (N), as described by Thompson, *et al.* (2002):

$$\Delta x = \sqrt{\frac{s^2 + \frac{a^2}{12}}{N} + \frac{8\pi s^4 b^2}{a^2 N^2}} \tag{1.2}$$

When background noise is low, the localization precision for a given pixel size and point-spread function is:

$$\Delta x \propto \frac{1}{\sqrt{N}} \tag{1.3}$$

Thus, to achieve the most precise localization, it is important to collect as many photons as possible. Indeed, Yildiz et al. (2003) achieved 1.5-nm resolution *in vitro* by collecting 5,000–10,000 photons per spot image.

By connecting localized points together in time to form single-molecule trajectories, molecular motions can be observed with nanometer precision. Yildiz et al. (2003) used single-molecule super-resolution trajectories to determine that myosin molecular motors "walk" in a hand-over-hand fashion, rather than according to the inchworm model. Ex-

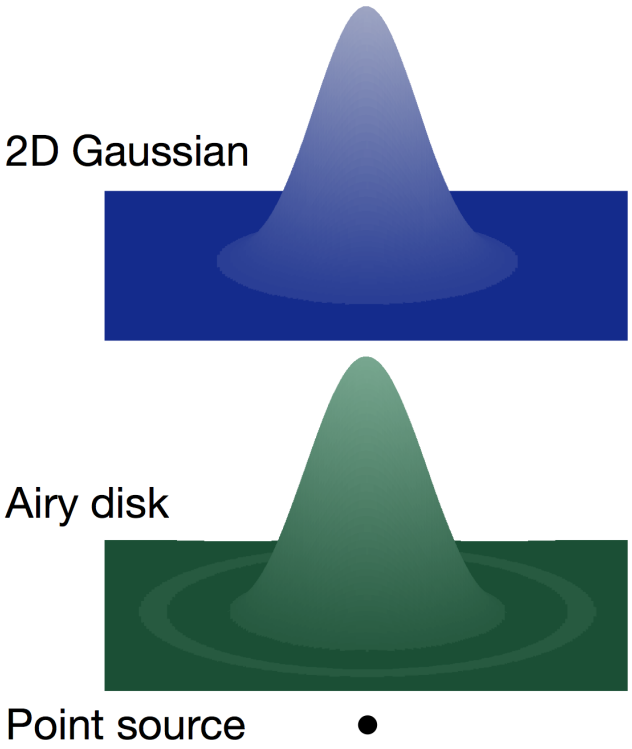

Figure 1.3: Comparison of an Airy disk and a two-dimensional Gaussian function
The diffraction-limited image of a point emitter is an Airy disk. The Gaussian function closely resembles the central lobe of the Airy disk and therefore is a reasonable approximation for fitting a point-spread function.

amples of single-molecule trajectory investigations in bacteria include the determination of diffusion coefficients for the *E. coli* membrane protein TatA (van den Wildenberg et al., 2011), and examination of the dynamics of the histamine protein kinase PleC and the structural protein MreB in *C. crescentus* (Deich et al., 2004; Kim et al., 2006).

This ability to extract position information from isolated fluorophores still requires two nearby fluorophores to be separated by the standard diffraction limit. Photoreactive fluorophores, which can undergo conformational changes to switch from dark, non-fluorescent states to fluorescent states, or from one emission wavelength to another (Subach et al., 2009b; Chudakov et al., 2010), enable high-resolution localization of densely labeled objects (Betzig et al., 2006; Rust et al., 2006; Hess et al., 2006) by separating fluorophores in time, rather than in space. By turning individual fluorophores on and off over several seconds or minutes (many imaging frames), the molecules can be distinguished separately in space with a high enough spatial frequency to satisfy the Nyquist–Shannon sampling theorem (Shannon, 1949; Shroff et al., 2008). Fitting of the point spread function leads to much improved resolutions from the ∼200 nm diffraction limit to tens of nanometers.

The inner workings of bacteria have been difficult to observe directly due to their small size. *V. cholerae*, the bacteria that is the focus of this thesis, is about 2 µm long and 0.6 µm in diameter, yet it has killed millions with its potent toxin (Kay et al., 1994; Centers for Disease Control and Prevention, 2013). We examine the pathogenic pathway of this disease-causing microbe using single-molecule super-resolution microscopy.

Single-molecule imaging methods as applied to live bacteria will be discussed in greater detail in Chapter 2, with examples drawn from our work in *V. cholerae* and from other systems described in the literature. By applying these methods, we have imaged proteins in *V. cholerae* with better than 40 nm precision, tracked diffusion as fast as 0.035 $\mu m^2/s$, and advanced our understanding of intracellular dynamics in the virulence regulation pathway of a human pathogen.

1.3 Cholera

The human disease cholera is caused by the Gram-negative bacterium *V. cholerae*, which is endemic to India and Bangladesh and present in most of the world (Kaysner and Hill, 1994). Cholera has been known since ancient times, but the disease remains a threat to human health in the developing world to the present day. There have been seven pandemics in the last two centuries (Fig. 1.4; Kaysner and Hill, 1994).

Sanitation infrastructure is very important for preventing the spread of cholera. *V. cholerae* spread through water contaminated with fecal matter from infected individuals. Chlorination of water supplies is effective at killing these bacteria, but in poor regions of the world, such treatment can be prohibitively expensive (Kaysner and Hill, 1994). *V. cholerae* are facultative anaerobes (Kay et al., 1994); they colonize the small intestine, but they can

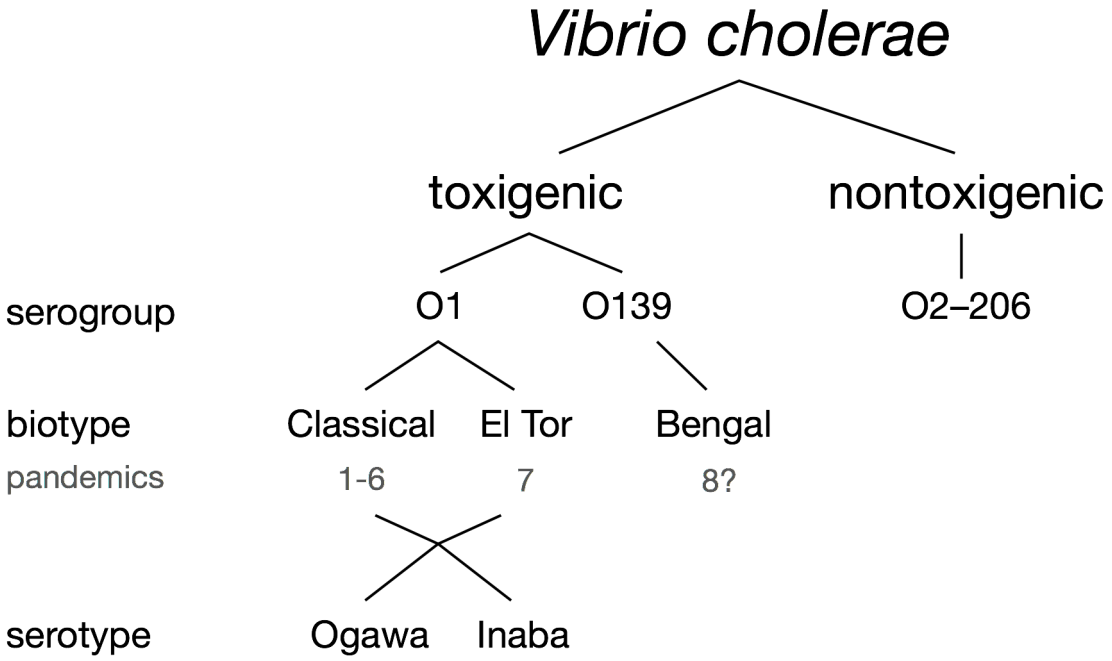

Figure 1.4: Categorization of *V. cholerae* strains
Cholera strains are categorized by differences in their surface antigens. The first six pandemics are believed to be caused by "classical" cholera in serogroup O1 (Kaysner and Hill, 1994). The seventh, ongoing cholera pandemic is caused by a different O1 strain, called El Tor. A possible eighth pandemic may be caused by a strain called Bengal, which belongs to serogroup O139 (Morris, Jr., 1994). Other serogroups of cholera exist, but do not produce cholera toxin (Colwell and Huq, 1994). In this thesis, we study TcpP motion in classical cholera strains.

also survive outside their hosts in saline and estuarine environments for months or more (Colwell and Huq, 1994).

In 1884, Robert Koch noted that *V. cholerae* produces a 'special poison' (Kaper et al., 1994). Yet it was 75 years before the presence of cholera toxin (CTX) was confirmed, and longer before the toxin protein was purified (De, 1959; Hall, 2011). The three-dimensional crystal structure of the protein complex was published in 1995 (Kaper et al., 1994; Zhang et al., 1995).

When an individual consumes food or water contaminated with *V. cholerae*, the bacteria that survive the passage through the stomach colonize the small intestine and begin producing CTX (Matson et al., 2007). The CTX protein is formed from two subunits, CtxA and CtxB, in a 1:5 ratio. This protein complex is absorbed into the mucosal cells, triggering up-regulation of cAMP, and leading to hypersecretion of salts and water: the characteristic "rice water" diarrheal symptom of the disease (Rodrigue et al., 1994). If left untreated, cholera is fatal to more than 50% of infected individuals within 24 hours (Bennish, 1994). The disease is treated primarily through its symptoms: oral or intravenous rehydration therapy is the most common treatment (Morris, Jr., 1994). Antibacterials may also be used, though many pathogenic *V. cholerae* strains are resistant to common treatments. For example, the primary *V. cholerae* strain present in Haiti since 2010 is resistant to several antibacterial agents, including streptomycin and sulfisoxazole (Cravioto et al., 2011). 70% of serogroup O1 isolates (i.e. the majority of *V. cholerae* strains that produce cholera toxin) were already tetracycline-resistant twenty years ago (Morris, Jr., 1994). Two oral vaccines are available for cholera, but they offer incomplete protection against the disease (CDC, 2013).

Biochemical evidence indicates that production of CtxAB and the toxin co-regulated pilus proteins (TcpA-F) are regulated by the cytoplasmically soluble protein ToxT, the primary direct transcription activator in cholera pathogenicity (Fig. 1.5; Beck et al., 2004; Withey and DiRita, 2006). ToxT in turn is controlled by the ToxR regulon which is composed of four bitopic inner-membrane-bound proteins: ToxR, TcpP, ToxS and TcpH (DiRita and Mekalanos, 1991; Krukonis et al., 2000; Beck et al., 2004; Withey and DiRita, 2006). TcpP is required for activation of *toxT* transcription, while ToxR enhances the effect of TcpP (Häse and Mekalanos, 1998; Krukonis et al., 2000; Matson et al., 2007). Overexpression of TcpP can compensate for the absence of ToxR, but the reverse is not true (Higgins and DiRita, 1994; Häse and Mekalanos, 1998; Murley et al., 1999; Krukonis et al., 2000). Both ToxR and TcpP have DNA-binding regions at their cytoplasmic N-termini (Miller et al., 1987; Häse and Mekalanos, 1998). The predominantly periplasmic proteins ToxS and TcpH are hypothesized to interact with ToxR and TcpP at their periplasmic C-termini in a stabilizing manner (DiRita and Mekalanos, 1991; Beck et al., 2004).

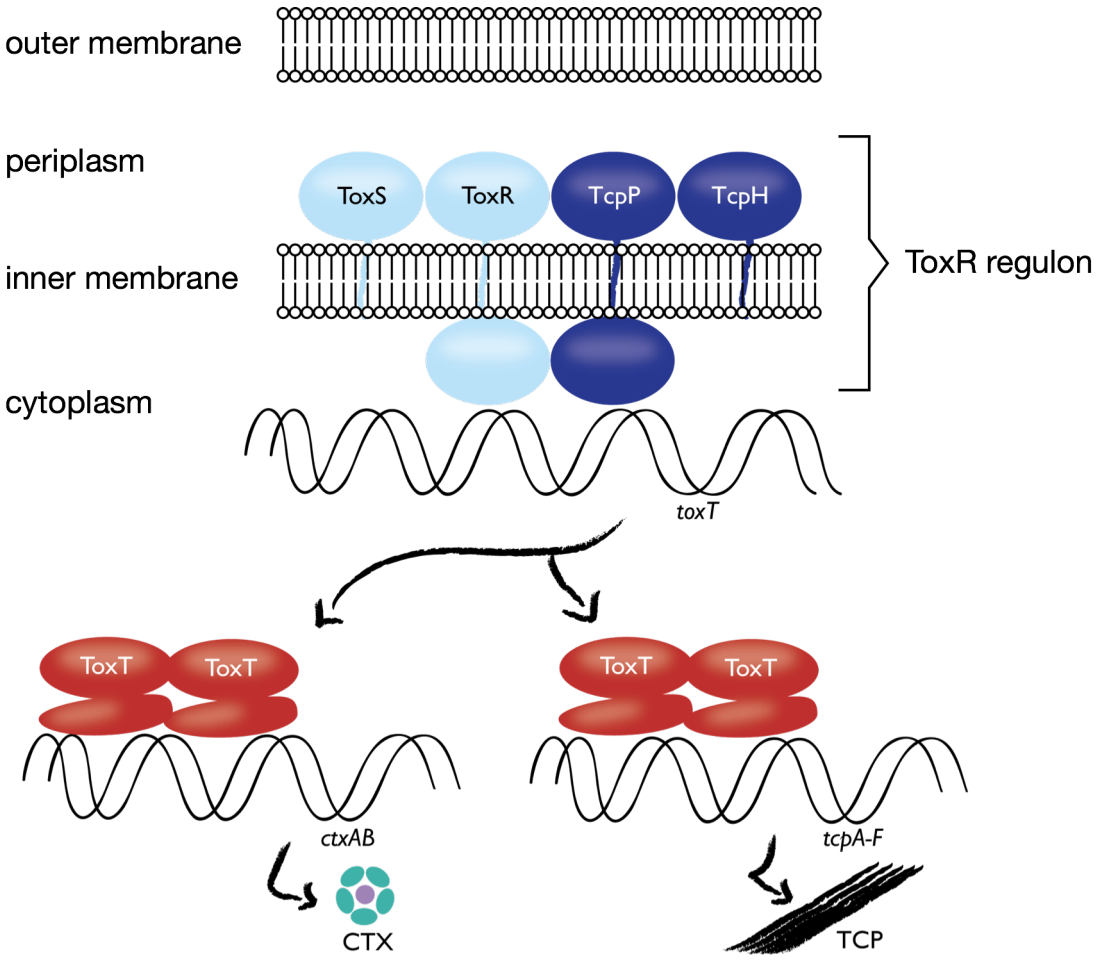

Figure 1.5: The ToxR regulon and the *V. cholerae* pathogenic pathway.
TcpP and ToxR bind DNA at the *toxT* promoter, and activate transcription of *toxT*. ToxS and TcpH stabilize ToxR and TcpP, respectively. ToxT activates transcription of *ctxAB* and *tcpA–F*.
Figure modified from Matson et al., 2007.

Table 1.1: Membrane-bound transcription activators
Several bacteria posses membrane-bound transcription activators. Membrane-bound transcription is also present in the archaea *Sulfolobus acidocaldarius*.

Species	Protein(s)	References
Vibrio cholerae	ToxR	Miller et al., 1987
	TcpP	Häse and Mekalanos, 1998
	CadC	Merrell and Camilli, 2000
Vibrio fischeri	LuxR	Kolibachuk and Greenberg, 1993
	ToxR	Reich and Schoolnik, 1994
Vibrio parahaemolyticus	ToxR	Lin et al., 1993
Escherichia coli	CadC	Neely et al., 1994
Bacteroides thetaiotaomicron	SusR	D'Elia and Salyers, 1996
Yersinia pseudotuberculosis	PsaE	Yang and Isberg, 1997
Photobacterium spp.	ToxR	Welch and Bartlett, 1998
Salmonella typhimurium	MarT	Blanc-Potard et al., 1999
Sulfolobus acidocaldarius	ArnR	Lassak et al., 2013

In eukaryotes, DNA is usually confined within the nucleus, and though prokaryotic DNA resides in the cytoplasm, it is mostly compacted in the cell center (Dame, 2005). Typical transcription activators can move through the cytoplasm using facilitated diffusion to find their promoters (Berg et al., 1981). TcpP and ToxR, restricted to the inner membrane, cannot search for *toxT* in the same way, alternating between one-dimensional and three-dimensional motions. Instead, the membrane limits their motion to two dimensions. TcpP and ToxR are uncommon in this regard, but not unique. A handful of other membrane-bound transcription activators are known (Table 1.1). The diffusion of the transcription activator TcpP and its interactions with ToxR and the *toxT* promoter are investigated in Chapter 3. A possible mechanism for this unusual membrane-bound transcription process is also provided.

1.4 Plasmon-enhanced fluorescence

As described in Section 1.2, localization precision in single-molecule imaging primarily depends on the number of photons detected. The photon count can be improved in several ways: a brighter or more stable fluorophore can be attached to the protein of interest, a larger NA objective can be used, the excitation power can be increased, or the integration time can be lengthened. Each of these approaches to improving resolution have advantages and disadvantages. Organic dye molecules typically used for fluorescence imaging have larger quantum yields and emit more photons before photobleaching than fluorescent proteins (Table 1.2; Tsien, 1998; Shaner et al., 2004; Willets et al., 2005), but they have several drawbacks as fluorophores in live bacteria cell imaging, as described in

Table 1.2: Comparison of fluorescent proteins with a small molecule dye
QY = quantum yield, Photons = total photons emitted before photobleaching. Rhodamine 6G is a small molecule dye. GFP and mCherry are fluorescent proteins.

Name	QY	Photons	References
rhodamine 6G	0.95	$\sim 10^6$	Willets et al., 2005
GFP	0.79	$\sim 10^5$	Tsien, 1998; Kubitscheck et al., 2000
mCherry	0.22	–	Shaner et al., 2004

Chapter 2. The specificity and relative ease of labeling, as well as the lack of special buffer requirements, make fluorescent proteins the preferred choice for fluorescence imaging in live bacteria. There is also a limit to an objective's numerical aperture, though multi-objective setups have been used to image fixed eukaryotic cells at resolutions better than the diffraction limit (Xu et al., 2012). Increasing the excitation power generates more photons per frame, but shortens the time until photobleaching. Longer imaging frames also have more photons per molecule and increase the signal-to-noise ratio for static fluorophores, but fast-moving molecules are more blurred (Zareh et al., 2012). We propose to use plasmon-enhanced electromagnetic fields, which are commonly used in surface-enhanced Raman spectroscopy (SERS) and other sensing applications (Ruemmele et al., 2013), to increase the photon count without sacrificing photobleaching lifetime or decreasing the signal-to-noise ratio.

A localized surface plasmon is the collective oscillation of free electrons; this resonant mode can be supported by metal nanoparticles (Orfanides et al., 2000). The size, shape and material of the nanoparticle determine the plasmon resonance frequency. The interaction of the nanoparticle with light at this same frequency excites the plasmon, creating an enhanced electric field that extends 20–50 nm from the nanoparticle edge (Anger et al., 2006). This field will change the local environment of a nearby fluorophore, and, in the right conditions, the metal nanoparticle can act as an optical antenna, enhancing the absorption and emission of the fluorophore, leading to longer photobleaching lifetimes and increased photostability (Sundaramurthy et al., 2004; Willets and Van Duyne, 2007; Taminiau et al., 2008). Both of these features are advantages for super-resolution imaging and may greatly improve localization accuracies for molecules in high-background samples like live cells.

The plasmon enhancement effect is distance-dependent, with a maximum enhancement around 5 nm, quenching at shorter distances, and insignificant enhancement beyond 100 nm (Fig. 1.6; Anger et al., 2006). For single-molecule experiments directly on plasmonic surfaces, a spacer layer may be required to prevent direct contact between the fluorophore and the surface, which would lead to charge transfer and fluorescence quenching (Jose et al., 2013). For cells, however, such a spacer layer may not be necessary, since the cellular envelope can act as a natural separator. For *V. cholerae* cells, the cell enve-

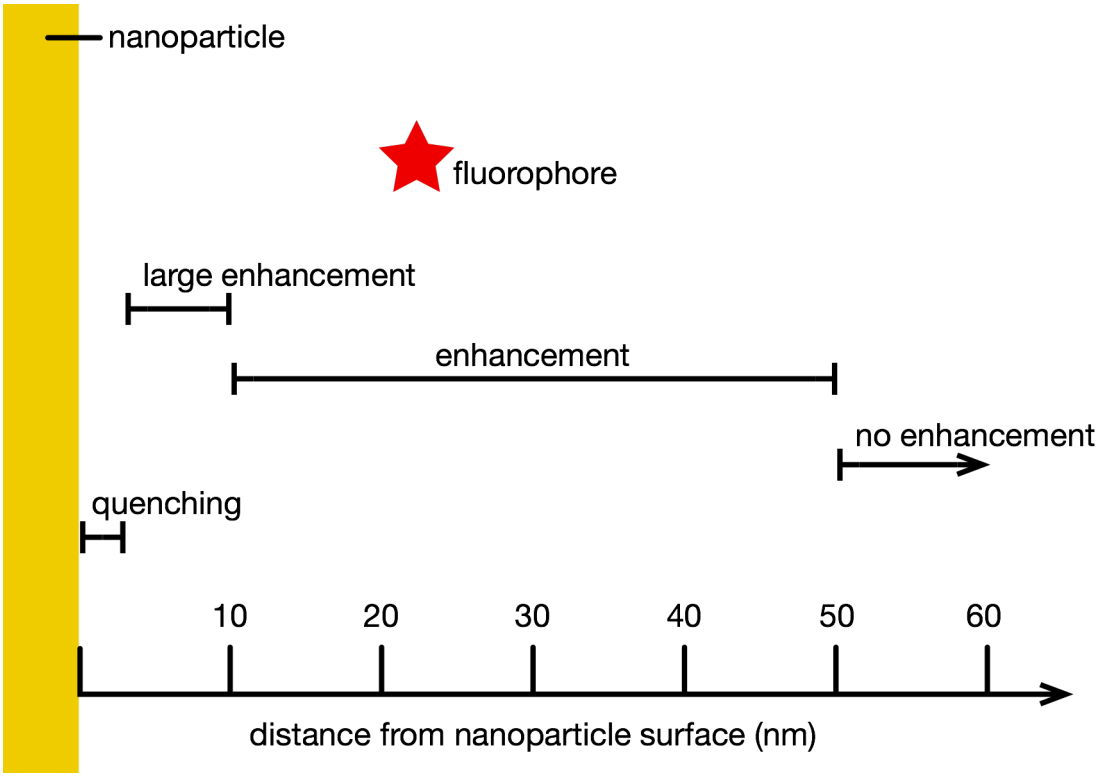

Figure 1.6: Plasmon-enhancement distance dependence
Maximum enhancement occurs 5–10 nm from the nanoparticle surface. Fluorophores closer than 5 nm may be quenched. Beyond 50 nm, no significant enhancement is observed (Anger et al., 2006).

lope is <50 nm thick (Graham et al., 1991), and so even inner-membrane-bound proteins should be near enough to experience the locally enhanced field around an extracellular gold nanoparticle.

We have developed a protocol for coupling photoactivatable TcpP-PAmCherry molecules on the inner membrane of live *V. cholerae* cells to extracellular patterned gold plasmonic substrates, which we present in Chapter 4.

1.5 Outline of thesis

In this thesis, we use fluorescence microscopy to investigate the nanometer-scale motions and interactions of membrane-bound proteins at the single-molecule level in live bacteria.

In Chapter 2, we discuss the methods developed and obstacles encountered in our work on TcpP-mCherry, TcpP-Dendra2, and TcpP-PAmCherry in live *V. cholerae*. We put our experiences in the context of the live-cell super-resolution microscopy literature in general.

In Chapter 3, we use super-resolution microscopy and single-molecule tracking to explore the dynamics of TcpP labeled with the photoactivatable fluorescent protein PAmCherry in live *V. cholerae* bacteria. We compare the motion of TcpP in three strains:

one very similar to wild-type *V. cholerae*, one lacking ToxR, and one lacking the *toxT* promoter to which TcpP and ToxR bind. We find that TcpP diffuses faster in the presence of both of its binding partners than when either is missing. Using mean squared displacement and cumulative probability distribution analyses (Anderson et al., 1992; Schütz et al., 1997), we find three populations of TcpP molecules in each strain: a fast population, a slow population, and a population immobile within our 40-nm resolution. Our data support a mechanism of TcpP–ToxR–*toxT* interaction in which ToxR locates and binds the *toxT* promoter, then recruits TcpP to activate transcription.

In Chapter 4, we improve the resolution of localization microscopy in live and chemically fixed *V. cholerae* by coupling fluorophore emission to extracellular nanoscale patterned gold surfaces capable of generating plasmon-enhanced fields for increased fluorescence. We find that, though fluorescence can be detected above the background scatter of annealed gold films, the bacteria are more readily observed on nanotriangle arrays produced using nanosphere lithography.

In Chapter 5, we summarize our findings and consider their implications as well as future directions.

Chapter 2

Obstacles to optimizing dynamic live-cell super-resolution fluorescence experiments

2.1 Introduction

Single-molecule super-resolution fluorescence microscopy enables the investigation of biological questions within living cells at millisecond and nanometer-scale resolution. Existing reviews of single-molecule microscopy (e.g. Xie et al., 2008; Huang et al., 2009; Biteen and Moerner, 2010; van den Wildenberg et al., 2011) cover a variety of methods and applications for specialists in the field. Optimizing a new single-molecule experiment can be challenging, however. In this chapter, we summarize common obstacles to live-cell single-molecule microscopy and describe methods we have used and developed to overcome these challenges in live bacteria.

2.2 Beyond the diffraction limit

The wave nature of light limits the resolution achievable with a traditional light microscope (Abbe, 1873). This diffraction limit is approximately 200 nm for visible wavelengths when using microscope objectives with large numerical apertures. Objects and patterns smaller than this limit are unresolvable and appear blurred due to the diffraction limit of light. Even large protein molecules are well below this 200-nm limit, making direct observations of protein conformations and interactions difficult (Fig. 2.1).

The small size of bacteria (generally 1–10 µm long; Koch, 1996) present an additional challenge. A *Vibrio cholerae* bacterium, for example, is only 10 times as long as the diffraction limit, making sub-cellular resolution a challenge. Bacteria also lack most of the organelles of eukaryotes, so cellular functions, from metabolism to transcription, are carried out in the cytoplasm (Koch, 1996). Cytoplasmically soluble proteins diffuse very

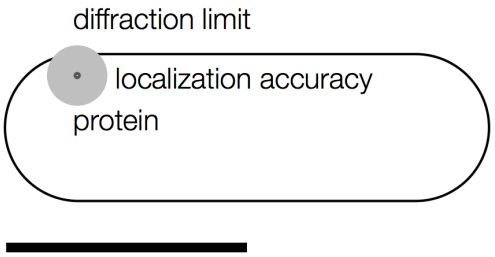

Figure 2.1: Size of a bacterium compared to the diffraction limit
The protein located at the center of the circle would appear blurred to the diffraction limit (∼300 nm, gray circle). A fluorophore could be localized to a greater position (∼30 nm, black circle). The scale bar represents 1 µm.

rapidly (the diffusion coefficient, D, is ~ 7 µm^2/s for enhanced yellow fluorescent protein in *Escherichia coli*; Kumar et al., 2010) in three dimensions, which makes them more difficult to track. Despite years of study, many questions remain about how cellular components find their targets or interaction partners in the crowded and dynamic environment of the bacterial cytoplasm. Even on the membrane, the small size of a bacterium produces a large degree of surface curvature, distorting observations on two-dimensional detectors (Deich et al., 2004; van den Wildenberg et al., 2011). Early bacterial microscopy studies focused on cell morphology (Rosselló-Mora and Amann, 2001), and staining methods, such as the Gram test, informed investigations into bacterial membrane structure (Baker and Bloom, 1948). Bulk fluorescence methods, such as Fluorescence Correlation Spectroscopy (FCS) and Fluorescence Recovery After Photobleaching (FRAP) have provided views of average motions, but a truly molecular-scale picture of live, dynamic cells is still quite new (Magde et al., 1972; Peters et al., 1974; van den Wildenberg et al., 2011). Even now, with single-molecule resolution imaging available, most live-cell studies of bacteria explore model systems (e.g. *E. coli*, *Caulobacter crescentus*; Xie et al., 2008; Biteen and Moerner, 2010; Wang et al., 2011; Bakshi et al., 2012).

In this chapter, we use examples from our work on the human pathogen *V. cholerae*, as well as studies from the literature on other systems, to illustrate the obstacles encountered when designing single-molecule microscopy experiments in live bacteria to push beyond the traditional diffraction limit. We will draw on our experiences investigating the diffusion of the membrane-bound transcription activator protein TcpP.

TcpP, along with three other proteins, is part of the ToxR regulon (Fig. 1.5), which binds DNA to activate transcription of *toxT*. The ToxT protein, in turn, activates transcription of cholera toxin (*ctxAB*) and the toxin co-regulated pilus (*tcpA-F*). In addition to its importance as the causative agent of the disease cholera, *V. cholerae* is interesting for its unusual biology: ToxR and TcpP are bound to the inner membrane of the bacterium, yet still manage to bind DNA (Miller et al., 1987; Häse and Mekalanos, 1998). Membrane-bound transcription is not unique to *V. cholerae*, but few other organisms

are known to posses this mechanism (Table 1.1). In our experiments, TcpP was labeled at its periplasmic C-terminus with a fluorescent protein, such that its diffusion could be tracked without disrupting DNA-binding at its cytoplasmic N-terminus (Häse and Mekalanos, 1998).

2.3 Super-resolution microscopy

To overcome the diffraction limit, super-resolution techniques have been developed; notably Fluorescence localization In One-nanometer Accuracy (FIONA) (Yildiz et al., 2003), Photoactivated Localization Microscopy (PALM) (Betzig et al., 2006), Fluorescence Photoactivation Localization Microscopy (FPALM) (Hess et al., 2006) and Stochastic Optical Reconstruction Microscopy (STORM) (Rust et al., 2006). Other super-resolution microscopy methods, such as Stimulated Emission Depletion (STED) and Structured Illumination Microscopy (SIM; See Table 2.1), employ patterned excitation, rather than localized single-molecule emission, and are reviewed elsewhere (Hell, 2007; Huang et al., 2009). In all of the single-molecule methods, target molecules that are labeled with fluorescent tags are observed by separating the emission of individual fluorophores in space or time, and the observed fluorescence intensity profiles are fit with a model function to localize each molecule with much greater precision than traditional light microscopy can achieve (Thompson et al., 2002). Today there are many variations of such localization microscopy. In addition to high-precision localizations, these methods enable tracking, three-dimensional imaging, and multicolor or multi-emitter detection (Table 2.1). Regardless of the configuration, each of these single-molecule microscopy experiments must be optimized for the specific goals of the investigation; this includes a careful choice of the fluorophore, the labeling scheme, and the imaging parameters.

2.4 Fluorescent labels

Sub-cellular investigations are not limited to intrinsically fluorescent molecules. In fact, a fluorescent label must be used in most fluorescence imaging studies. The localization precision of a molecule depends on the number of fluorescent photons collected (Thompson et al., 2002); the more photons observed, the greater the precision of the position measurement can be. Background signals—in the form of cellular autofluorescence, diffuse fluorescence from very fast molecules, excess or unwanted fluorophores incorporated into the cell, and camera noise—worsen this precision. A high signal-to-noise ratio is easily achievable for a bright, stationary fluorophore in a controlled environment, but in live-cell experiments, fluorophores may be dim, background fluorescence can be high, and the molecules of interest may move during observation (Bates et al., 2007; Biteen et al.,

Table 2.1: Single-molecule and super-resolution microscopy methods
Other microscopy methods used to image in bacteria or in live cells are also listed below.

Method	Full Name	Reference
FIONA	Fluorescence Imaging with One-Nanometer Accuracy	Yildiz et al., 2003
SHRImP	Single-molecule High-Resolution Imaging with Photobleaching	Gordon et al., 2004
NALMS	Nanometer-Localized single-Molecule fluorescence microscopy	Qu et al., 2004
SHREC	Single-molecule High-Resolution Colocalization	Churchman et al., 2005
PALM	Photoactivated Localization Microscopy	Betzig et al., 2006
STORM	Stochastic Optical Reconstruction Microscopy	Rust et al., 2006
FPALM	Fluorescence Photoactivation Localization Microscopy	Hess et al., 2006
PAINT	Point Accumulation for Imaging in Nanoscale Topography	Sharonov and Hochstrasser, 2006
PALMIRA	PALM with Independently Running Acquisition	Egner et al., 2007
dSTORM	Direct STORM	Heilemann et al., 2008
SOFI	Superresolution Optical Fluctuation Imaging	Dertinger et al., 2009
uPAINT	Universal PAINT	Giannone et al., 2010
CALM	Complementation Activated Localization Microscopy	Pinaud and Dahan, 2011
BALM	Binding-Activated Localization Microscopy	Schoen et al., 2011
SPRAIPAINT	Superresolution by Power-dependent Active Intermittency PAINT	Lew et al., 2011
BaLM	Bleaching/blinking assisted Localization Microscopy	Burnette et al., 2011
TALM	Tracking and Localization Microscopy	Appelhans et al., 2012
SMACM	Single-Molecule Active-Control Microscopy	Moerner, 2012
FRET	Förster Resonance Energy Transfer	Förster, 1948
FCS	Fluorescence Correlation Spectroscopy	Magde et al., 1972
FRAP	Fluorescence Recovery After Photobleaching	Peters et al., 1974
STED	Stimulated Emission Depletion	Hell and Wichmann, 1994
(S)SIM	Saturated Structured-Illumination Microscopy	Gustafsson, 2005

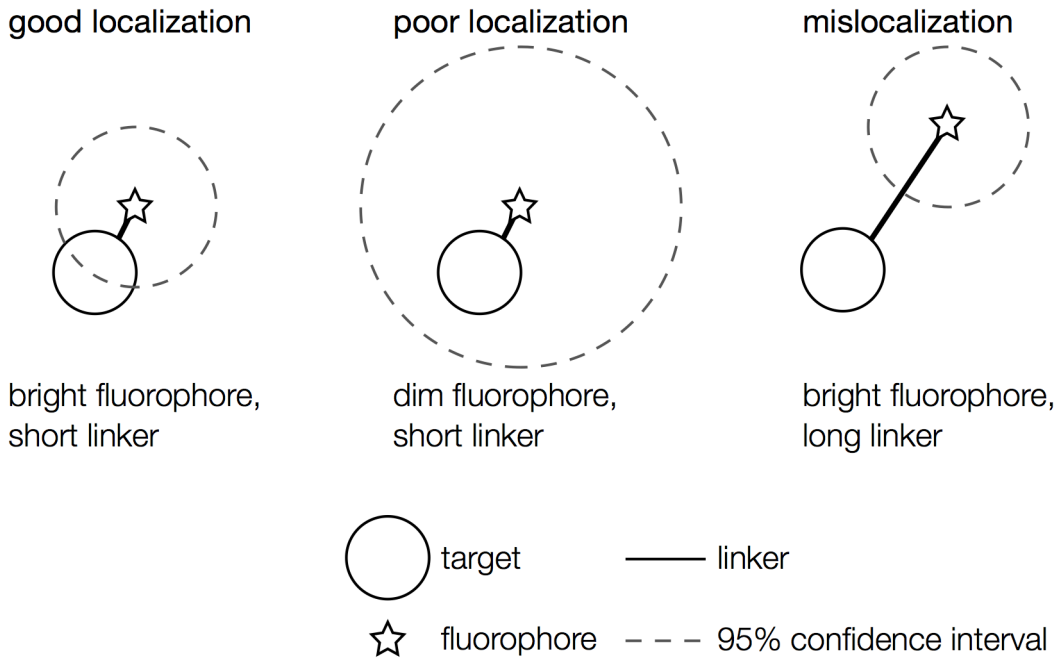

Figure 2.2: Localization problems arising from dim fluorophores and long linkers
A bright fluorophore that is near the feature of interest is well suited for localization. Dimmer fluorophores cannot be localized as well. A precisely localized fluorophore that is far from the feature it is intended to label does not report the desired position accurately.

2008; Rowland and Biteen, 2014). Fluorophore bleaching and blinking can also reduce detectability.

2.4.1 General considerations

There are several things to consider when choosing an appropriate fluorophore and labeling scheme for a particular experiment. Ideally, the fluorophore used should report the position of its target very precisely without perturbing the system in any way. If the linker between the fluorophore and its target is too long, as can be the case, for instance, for antibody labeling (Ritchie et al., 2013), mislocalizations can occur: the fluorophore may be localized with high precision, but if the probe is several nanometers from the point of interest, the uncertainty in the target position is much greater than the fluorophore localization accuracy suggests (Fig. 2.2). Additionally, large tags may hinder the diffusion of small, mobile molecules in the cytoplasm because the diffusion coefficient is inversely related to particle size (Einstein, 1905).

In addition to local effects, it is important to check that labeling the target molecule does not interfere with larger-scale cellular functions. Whenever possible, labeled cells should grow at similar rates to unlabeled cells and exhibit normal phenotypes. Controls

for downstream activity are also crucial. For example, in our studies of *V. cholerae* expressing TcpP-PAmCherry as the sole TcpP source, we found that the toxin coregulated pilus protein TcpA, whose expression is regulated by TcpP, is produced at levels comparable to wild-type expression. However, some effects of the TcpP-PAmCherry fusion were observed. For instance, *V. cholerae* expressing TcpP-PAmCherry does not require TcpH, the protein known to protect wild-type TcpP from degradation (Fig. 1.5; Beck et al., 2004). This indicates that the periplasmically localized fluorescent protein tag may mimic the stabilizing role of TcpH.

In two other examples of biological activity controls, Xie *et al.* found that *E. coli* T7 RNAP labeled with the yellow fluorescent protein Venus at the N-terminus maintains its polymerase activity, and that the protein Tsr maintains the ability to enter to membrane when labeled at its C-terminus (Yu et al., 2006; Xie et al., 2008). Additionally, labeling the *E. coli lac* repressor does not impede DNA-binding activity, but the tagged repressor forms dimers, rather than the tetramers present in the wild-type (Elf et al., 2007).

Due to these restrictions on size and function, the most common fluorophores in live-cell single-molecule localization microscopy experiments are fluorescent proteins and small molecule dyes attached using enzymatic labeling schemes, though other labels may also be used (Chen and Ting, 2005; Xie et al., 2008; Huang et al., 2009; Biteen and Moerner, 2010; Henriques et al., 2011; van den Wildenberg et al., 2011).

2.4.2 Fluorescent proteins

A wide variety of fluorescent proteins are now available, with adaptations and characteristics to suit an array of applications: they can be pH-stable or -sensitive, and fast- or slow-maturing (Fernández-Suárez and Ting, 2008; Chudakov et al., 2010). Some fluorescent proteins have even been engineered in split forms that fluoresce when the two halves are combined, for instance when two labeled proteins interact (Hu and Kerppola, 2003; Chudakov et al., 2010; Pinaud and Dahan, 2011). Fluorescent proteins can also be paired for Förster Resonance Energy Transfer (FRET) experiments, though the low FRET efficiency of such pairs limits their utility for single-molecule FRET (Akrap et al., 2010).

Fluorescent proteins come in many colors, from blue to far-red, and in monomeric, dimeric and tetrameric forms. The canonical green fluorescent protein GFP is naturally a monomer (Ormö et al., 1996; Chudakov et al., 2010), but many other fluorescent proteins, notably the red fluorophores derived from DsRed, which was isolated from coral (*Discosoma* spp.; Matz et al., 1999), rather than jellyfish (*Aequorea victoria*), have a tendency to oligomerize (Shaner et al., 2004), particularly at high concentrations. This may cause fluorescent protein-tagged molecules to cluster together in ways that unlabeled native molecules would not, giving rise to mislocalization artifacts (Verkhusha and

Lukyanov, 2004). Still, several red fluorescent proteins, such as mCherry, have been developed with improved monomeric character (Shaner et al., 2004). Additionally, if the fluorescent protein concentration is very low, even proteins that tend to dimerize will have a low probability of finding a partner to pair with.

Fluorescent proteins can also be photoactivatable, photoshiftable, or photoswitchable (Fig. 2.3; Fernández-Suárez and Ting, 2008). A photoactivatable fluorophore begins in a non-fluorescent "dark" state and can be switched to a fluorescent state using violet light ($\lambda = 350$–400 nm). An example of this is PAmCherry, which, after activation with violet light (405 nm), absorbs yellow-green light (excitation $\lambda_{\max} = 564$ nm) and emits red light (emission $\lambda_{\max} = 595$ nm; Subach et al., 2009b). A photoshiftable fluorophore has two fluorescent states and can switch from the shorter-wavelength state to the longer-wavelength state. For example, Dendra2 has a "green" state (excitation $\lambda_{\max} = 491$ nm, emission $\lambda_{\max} = 507$ nm) and a "red" state (excitation $\lambda_{\max} = 554$ nm, emission $\lambda_{\max} = 531$ nm) and switches from "green" to "red" with 405-nm light (Gurskaya et al., 2006). A photoswitchable fluorophore, such as Dronpa, can *reversibly* change from a fluorescent state to a dark state. For Dronpa, blue light (excitation $\lambda_{\max} = 503$ nm) can excite the fluorophore to emit green light (emission $\lambda_{\max} = 518$ nm), or intense 488-nm light can switch the molecule to a dim state. 405-nm light reactivates the fluorescent state (Habuchi et al., 2005).

Genetically encoding a fluorescent protein tag allows highly specific labeling of the target, but this advantage must be weighed against the poor quantum yields (e.g. 0.22 for mCherry) and large size (∼25 kDa) of these labels as compared to organic dyes, e.g. rhodamine 6G (QY = 0.95, ∼0.5 kDa, i.e. ∼500 g/mol; Shaner et al., 2004; Willets et al., 2005). Moreover, not all protein fusions are stable. For example, in *V. cholerae* bacteria expressing a fusion of the membrane-bound transcription activator protein TcpP to the photoshiftable fluorescent protein Dendra2, we found that the TcpP-Dendra2 fusion had a higher rate of degradation than native TcpP, though this fusion protein was still able to activate *toxT* transcription. In this same organism, other fluorescent proteins, such as mCherry and PAmCherry, formed fusions with TcpP that were much more stable and thus more suitable for imaging (Figs. 2.4 and 2.5).

The desirable properties of a label depend on the goals of the experiment. The mCherry fluorescent protein exhibited rapid blinking in our *V. cholerae* investigations, precluding single-molecule tracking of TcpP-mCherry, since tracking ended each time a molecule turned off, shortening the trajectories observed. This fluctuation in mCherry fluorescence was advantageous for creating PALM super-resolution images of the TcpP localizations, however. Based on observations of TcpP-mCherry under 561-nm excitation, we achieved single-molecule levels of fluorescence without photoactivation simply based on mCherry molecules blinking during imaging and demonstrated that the TcpP-mCherry fusion is localized to the cell membrane (Fig. 2.6). This blinking behavior may be an example of

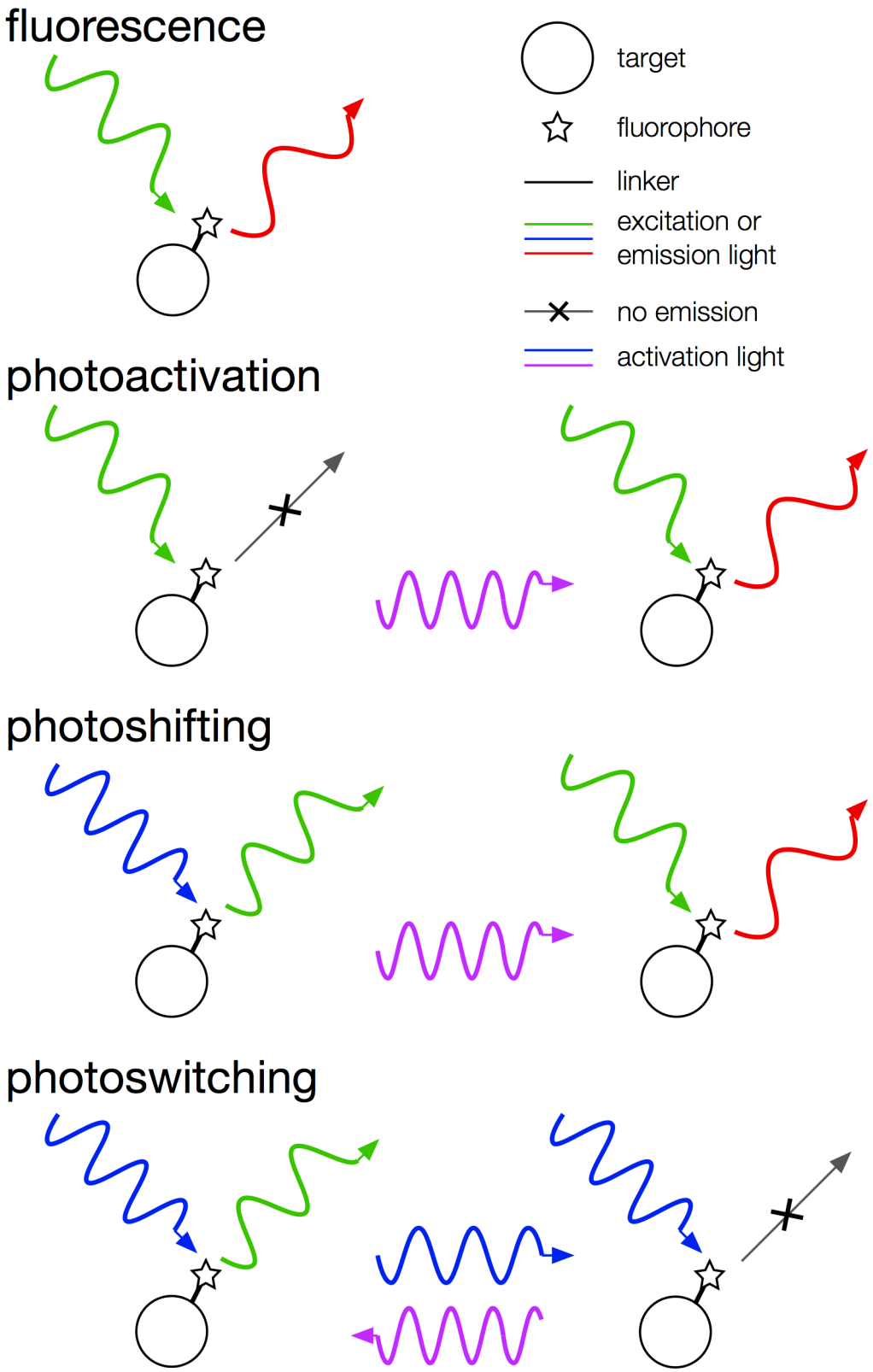

Figure 2.3: Fluorophore types
mCherry is an example of a typical fluorophore. PAmCherry is photoactivatable. Dendra2 is photoshiftable. Dronpa is reversibly photoswitchable.

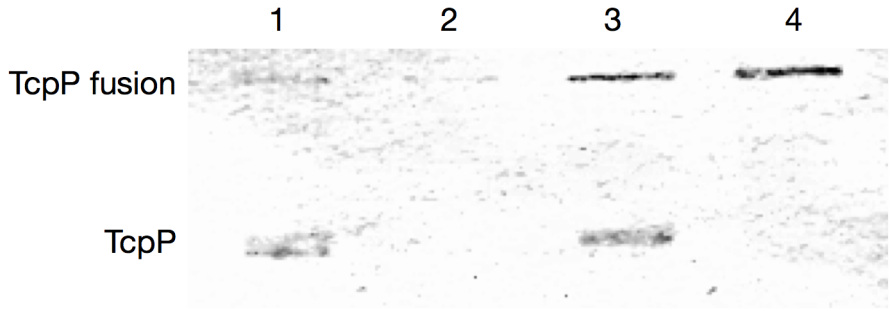

Figure 2.4: Immunoblotting with anti-TcpP serum
Lane 1: wild type O395 *V. cholerae*
Lane 2: Δ*tcpP*
Lane 3: Δ*tcpP* TcpP-mCherry
Lane 4: Δ*tcpP* TcpP-PAmCherry
In a Δ*tcpP* strain, TcpP is not expressed (lane 2). Insertion of a plasmid containing TcpP-mCherry or TcpP-PAmCherry restores expression of TcpP under inducing conditions (lanes 3 and 4).

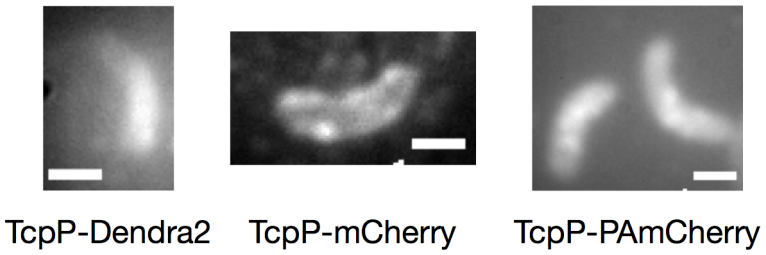

Figure 2.5: Diffraction limited fluorescence images of *V. cholerae* cells expressing TcpP-Dendra2, TcpP-mCherry and TcpP-PAmCherry
Each image is the sum of frames over several seconds. Cells expressing TcpP labeled with Dendra2 (left) are very difficult to detect above background. Bright foci can be seen in the cells with TcpP-mCherry and TcpP-PAmCherry (center, right). The scale bar is 1 µm in each panel.

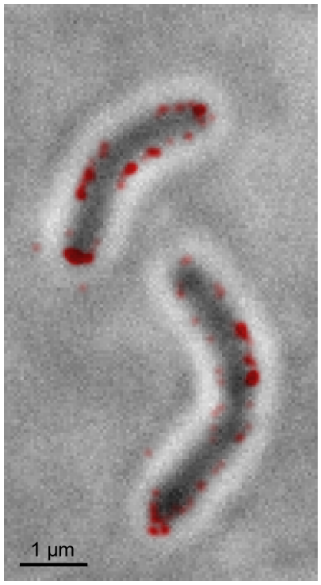

Figure 2.6: TcpP-mCherry localizes to the cell membrane
Red foci are super-resolution localizations blurred to the 95% confidence intervals of their fits. The localizations are plotted on top of a reverse-contrast bright field image of the cells.

temporary quenching, known as "kindling," which is known to occur in mCherry and a handful of other proteins at low excitation powers (Chudakov et al., 2003; Shaner et al., 2008).

PAmCherry fluorescence was more consistent than mCherry and had the added advantage that we could control its activation with careful doses of 405-nm laser illumination. On the other hand, PAmCherry requires a more complicated and more expensive optical setup: a second (activation) laser (405 nm) must be coaligned with the imaging laser (561 nm; Fig. 2.7). It has also been reported recently that only about 50% of PAmCherry molecules photoactivate into a fluorescent state (Durisic et al., 2014). Though this could not be measured in our experiments, we do not believe that inactive TcpP-PAmCherry affected our trajectory analysis, since we observe dozens of molecular trajectories per cell, and there should therefore be a sufficient number of activated molecules to represent the various modes of TcpP motion. Certainly, however, inactivatable fluorophores could lead to underestimates in molecule-counting experiments.

Another consideration when using fluorescent proteins to label cellular components is the maturation time. Before a protein can fluoresce, it must fold properly and undergo an oxidation reaction to form the chromophore. Typical maturation times are around 40 minutes, but depending on pH, temperature and the specific fluorescent protein, some may take multiple hours to mature (Baird et al., 2000; Shaner et al., 2004, 2008; Chudakov et al., 2010). On the other hand, particularly fast-maturing proteins, such as the yellow fluorescent protein Venus, take less than 10 minutes to mature (Nagai et al., 2002). It is important to note that the chromophore formation step in GFP and DsRed deriva-

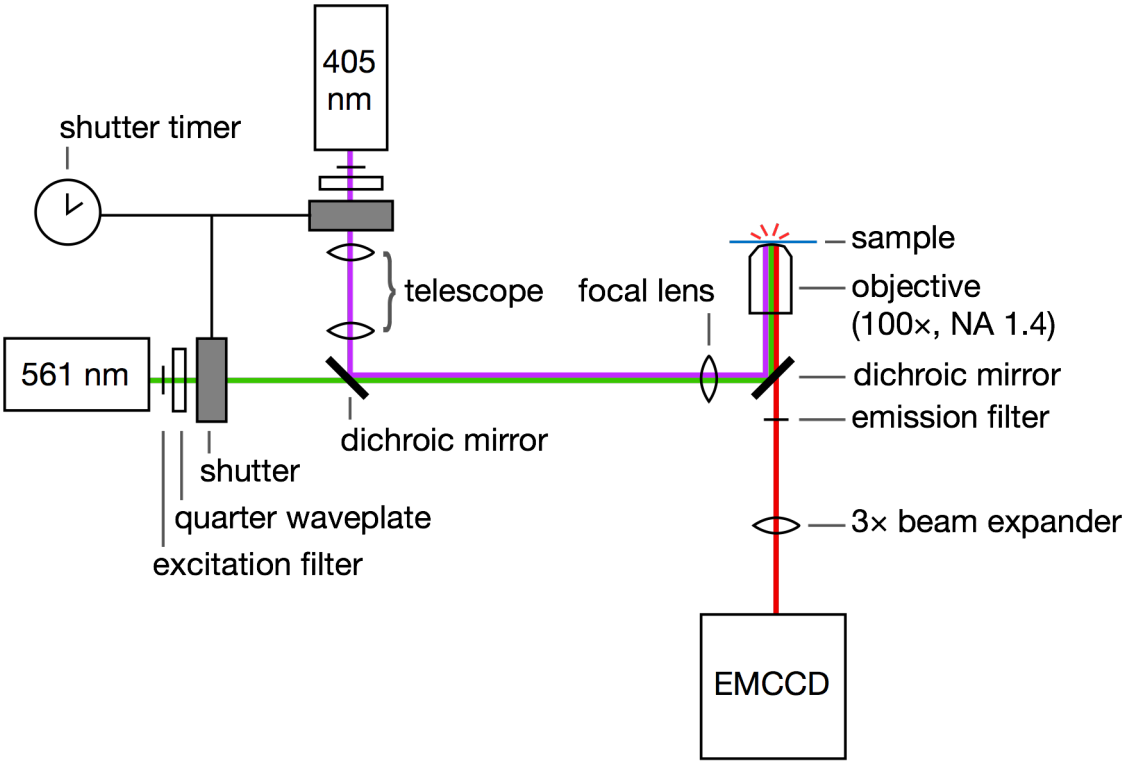

Figure 2.7: Laser setup
405- and 561-nm laser paths are coupled via dichroic mirrors. Fluorescence emitted from the sample is filtered using emission and dichroic filters and detected on an EMCCD camera.

tives requires the presence of oxygen, and therefore traditional fluorescent proteins are incompatible with strict anaerobes.

2.4.3 Small molecule dyes

Organic dye molecules have the advantage of being much brighter and more stable than fluorescent proteins and much smaller than biological molecules of interest. The simple addition of a small dye molecule is unlikely to hinder protein diffusion, and with many more photons emitted than from a fluorescent protein before photobleaching, the position of this probe can be determined with better precision. However, small molecule dyes are not genetically encoded and must be incorporated into the cell in some way. Not all dyes can permeate the membrane; rhodamine dyes generally can permeate bacterial membranes, but only poorly so; sulfonated cyanine dyes are unable to cross the membranes at all (Fernández-Suárez and Ting, 2008). Endocytosis and microinjection, which are used to introduce dyes to eukaryotic cells, are not available for bacterial cells. Membrane permeabilization must be used instead, though certain permeabilization methods cause artifacts (Sochacki et al., 2011). To achieve specific labeling of a protein with an organic dye, the protein must be engineered to incorporate a binding motif for the dye molecule to recognize and bind to covalently, such as in FlAsH or HaloTag, among others (Griffin et al., 1998; Fernández-Suárez and Ting, 2008; Los et al., 2008). In addition, since the free dye can bind nonspecifically elsewhere in the cell, most dye-labeling schemes are limited in their specificity, and all must be accompanied by washing steps to remove excess, unbound dye from the cell (Fernández-Suárez and Ting, 2008). Very few organic dyes are photoactivatable, as well. In fixed-cell imaging, blinking buffers can produce photoswitching, but most such buffers are cytotoxic (Henriques et al., 2011; Endesfelder et al., 2011).

Dye labeling schemes have been used for single-molecule microscopy in eukaryotes (Fernández-Suárez and Ting, 2008), but due to the difficulty transporting dyes inside the cells, there is little comparable work in bacteria. Bacterial extracellular membrane proteins have been labeled with dyes, though, through enzymatic or antibody labeling (Karunatilaka et al., 2014).

Dye molecules can also be used to label other biologically relevant molecules of interest, not just proteins. For example, in investigations of the interaction between the carbohydrate amylopectin and the outer membrane starch utilization system (Sus) proteins responsible for capturing and degrading this carbohydrate in the gut symbiont *Bacteroides thetaiotaomicron*, amylopectin was labeled with AlexaFluor 488 and the protein SusG was fused to the Halo enzyme and labeled with a fluorescent HaloTag (Los et al., 2008; Karunatilaka et al., 2014). Antibodies interfere with the desired starch–Sus protein interaction, so a small-molecule dye is well-suited to this case.

For fixed-cell studies, fluorescently-tagged antibody labels may be used (Fernández-Suárez and Ting, 2008). Antibodies can target specific proteins. This method can complement live-cell studies, for example, to check the specificity of another labeling scheme (Karunatilaka et al., 2014). However, antibodies are very bulky linkers, particularly when secondary antibodies are used. The fluorophore may be tens of nanometers away from the point of interest, adding to the uncertainty in the target's position. Antibodies may also bind to multiple molecules, further adding to the uncertainty in the position of the molecule of interest (Fernández-Suárez and Ting, 2008). Despite these disadvantages, antibody labeling of bacterial surfaces or fixed cells can be a worthwhile addition to live-cell single-molecule microscopy, validating the localizations observed with other methods.

2.4.4 Other labeling schemes

Several alternative labels and labeling schemes exist, as well, including the incorporation of unnatural amino acids (UAAs), and quantum dot or nanoparticle labeling. One avenue that is currently being pursued for single-molecule imaging in live bacteria is the incorporation of UAAs (Charbon et al., 2011). Like fluorescent protein fusions, UAAs can be highly specific, genetically encodable handles. In addition to encoding fluorescent UAAs, it is possible to incorporate a wide variety of functional groups, including ketone, azide and alkyne ligands (Chin et al., 2002; Deiters et al., 2003; Zhang et al., 2003; Chen and Ting, 2005), to enable specific, covalent attachment of organic dyes to proteins of interest. Due to their small size, UAAs can also be minimally perturbative, though proteins may be truncated at the modification site, rather than incorporating the desired unnatural amino acid (Chen and Ting, 2005).

Quantum dots and nanoparticles have also been used for labeling. Because of their size (2–10 nm; Mutavdžić et al., 2011), they are difficult to get into cells without endocytosis, so in bacteria they are generally restricted to the outer membrane and cell surface (Chalmers et al., 2007; Ritchie et al., 2013). Because diffusion is inversely related to particle size, nano-scale labels may slow the dynamics of their target (Einstein, 1905; Ritchie et al., 2013). Quantum dots and nanoparticles have much longer photobleaching lifetimes than fluorescent proteins and small molecule dyes, allowing longer-term observations, though they are also known to blink (Michalet et al., 2005; Mahler et al., 2008).

2.5 Sample considerations

2.5.1 Achieving single-molecule levels of fluorescence

Because single-molecule localization depends on detecting the emission from isolated emitters, most fitting algorithms can handle only one molecule per diffraction-limited

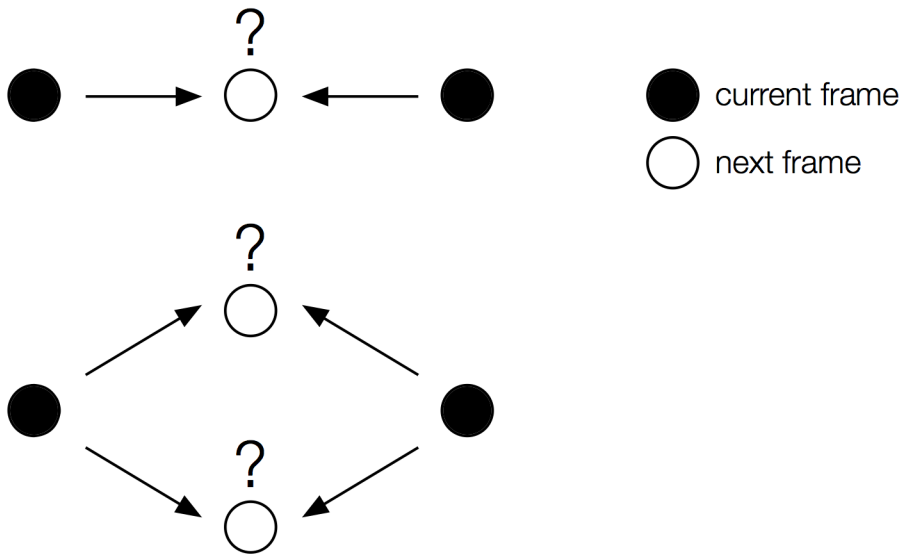

Figure 2.8: Difficulties with tracks crossing at high fluorophore densities
If too many localization occur in proximity to one another, the tracking algorithm may not be able to tell which molecule should follow which trajectory.

area ($\sim$0.25 µm^2) at a time. Thus, if the fluorophore density is too high, individual molecules cannot be resolved. For bacteria like *V. cholerae*, which are about 2 µm long and 0.6 µm in diameter (Kay et al., 1994), this limit is about five fluorophores at a time per cell in ideal conditions. Data processing schemes, including multi-fluorophore fitting and successive frame subtraction (e.g. Single-molecule High-Resolution Imaging with Photobleaching, or SHRImP; Gordon et al., 2004; Huang et al., 2011), are being developed to improve single-molecule detectability at higher labeling densities. Low fluorophore density may be desirable even when multi-emitter fitting is possible, though, if one is interested in diffusion dynamics. As fluorophore density increases, the likelihood of molecular trajectories intersecting increases, complicating single-molecule tracking and diffusion coefficient calculations (Fig. 2.8).

For a genetically-encoded tag, there are several ways to control fluorophore density. In some cases, the protein of interest has a low native expression level. Ectopically expressed target proteins can also be restricted to low copy numbers by controlling promoter induction or repression. For example, in studies of labeled ToxR in *V. cholerae*, our lab has found that the ToxR-fluorescent protein fusion is expressed at levels too high for single-molecule imaging, but the addition of glucose can repress gene expression enough to achieve single-molecule concentrations. On the other hand, even when many fusion proteins are expressed in a single bacterial cell, partial bleaching of the sample may decrease the concentration of active fluorescent proteins sufficiently to achieve single-molecule levels of fluorescence. To attain greater control, photoactivation, photoswitching, or photoshifting of the fluorescent proteins allows small subsets of these fluorophores to fluoresce

at the detection wavelength at a given time based on excitation by an activation laser (see Section 2.4.2).

In Δ*tcpP V. cholerae*, we express 10–40 copies of TcpP-PAmCherry from an arabinose-inducible promoter by incubating these mutant cells in 0.10% arabinose for 3 hours at 30 °C. The photoactivatable PAmCherry is initially dark, and a 70-ms 0.006–0.2 μW/μm^2 dose of 405-nm laser light renders 1–4 PAmCherry molecules fluorescent per cell. As discussed above, this activation laser can be circumvented entirely in the case of blinking fluorescent proteins. For example, when we express 10–20 copies of TcpP-mCherry in Δ*tcpP V. cholerae* under the same conditions as described for TcpP-PAmCherry above, the cells initially have too many fluorescing molecules for single-molecule imaging. After 2–3 minutes of photobleaching with a 0.2–0.3 μW/μm^2 561-nm laser, only about eight fluorescent mCherry molecules remain and the blinking dynamics of these probes during imaging ensures that 1–4 mCherry molecules are emissive in any single imaging frame.

2.5.2 Minimizing cell stress

Live cells pose additional challenges for single-molecule imaging as the integrity of the samples must be assured at all times. To represent the behavior of unlabeled molecules in a meaningful way, it is critical to prevent unnecessary stress to the cells. First, it is important to keep the cells in an appropriate extracellular environment: well hydrated with the necessary nutrients for continued growth and function. Additionally, the aerobic or anaerobic requirements of each particular organism must be considered. To image live obligate anaerobes like *B. thetaiotaomicron*, we have found it necessary to deaerate the buffer, add reducing agents such as cysteine, and seal our samples with epoxy before removal from the anaerobic growth chamber (Karunatilaka et al., 2014). At the same time, motile cells must be immobilized to provide a stationary frame of reference for single-molecule imaging. Preparing 1–2% agarose in cell media yields a gel with 100–200 nm pores (Narayanan et al., 2006), which leads to a surface roughness sufficient to prevent cells deposited on these agarose pads from moving during data capture while also maintaining a moist environment.

For *V. cholerae* experiments, we found that 2% wt/vol agarose in M9 minimal medium was sufficient to immobilize the bacteria for live cell imaging. Minimal media are preferred because rich media, such as LB, may add background fluorescence and increase cellular autofluorescence (Fig. 2.9), though we have found that single-molecule imaging is still possible in bacteria grown in LB, and indeed that this nutrient-rich media is sometimes preferred because it reduces cell stress. For imaging experiments lasting longer than an hour, samples may be sealed, for example using hot paraffin wax or epoxy, to prevent desiccation of the agarose pad, though sample sealing reduces oxygen delivery and is

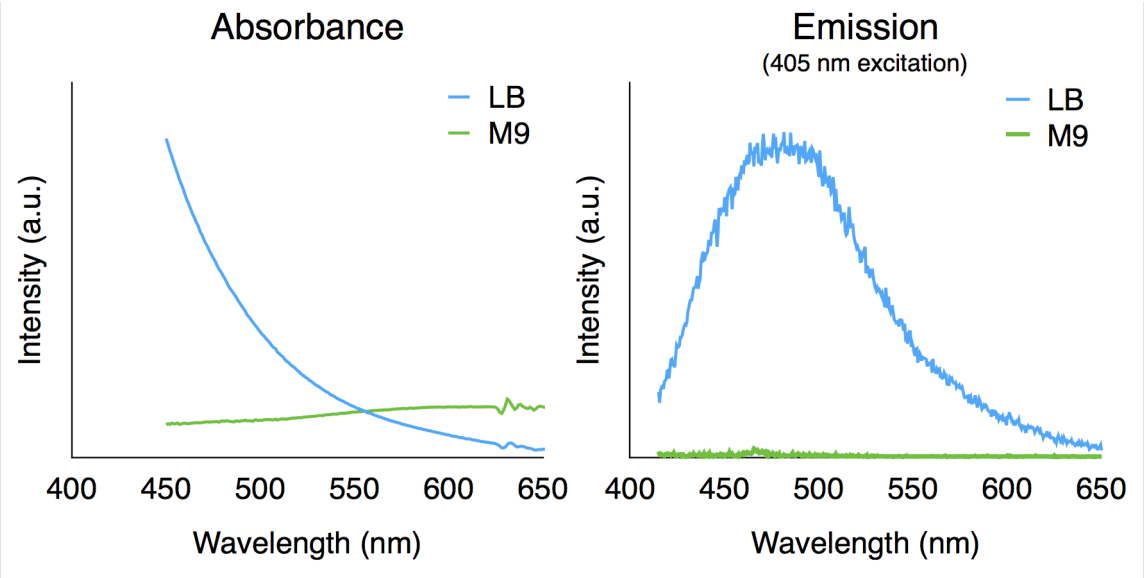

Figure 2.9: Absorption spectra and fluorescence emission spectra of LB and M9 media Fluorescence emission spectrum is under 405-nm illumination. Both LB and M9 have negligible fluorescence under 561-nm illumination.

therefore harmful for obligate aerobes (Fig. 2.10). A short (5–10 min) air-drying step may also be necessary before sealing to ensure that samples are not too wet. We have found that *V. cholerae* bacteria in samples sealed without this drying step continued moving and therefore could not be imaged despite the agarose pad surface roughness. In particularly wet samples, *V. cholerae* cells could be seen swimming rapidly in all directions or spinning in place, presumably as their flagella were caught on the agarose surface.

Alternatively, microfluidic devices may be used to hold cells in place and keep them nourished for longer or more complex experiments (Xiao et al., 2008; Moolman et al., 2013), and algorithms for handling single-molecule imaging within moving cells are being developed (Rowland and Biteen, 2014).

The activation and excitation lasers may also cause cell stress. Because cells are highly absorbent at the violet end of the spectrum, when photoactivating fluorophores in live cells using UV or violet light (λ = 350–450 nm), one must choose laser powers and exposure times carefully to avoid damaging the cells. At high power densities, even visible light radiation can generate reactive species that can damage a cell and lead to death (Xiao et al., 2008; Ano, 2013; Endesfelder and Heilemann, 2014). Wagner et al. (2010) found that 633 nm excitation at 400 J/cm^2 (corresponding to 0.4 µW/µm^2 in 100 ms) was sufficient to impact colony formation of U373-MG glioblastoma cells. Emission from fluorescent proteins themselves has a much lower power than the laser irradiation and is therefore a negligible source of cell stress. In our experiments imaging proteins in *V. cholerae*, we used low power densities for both activation and excitation

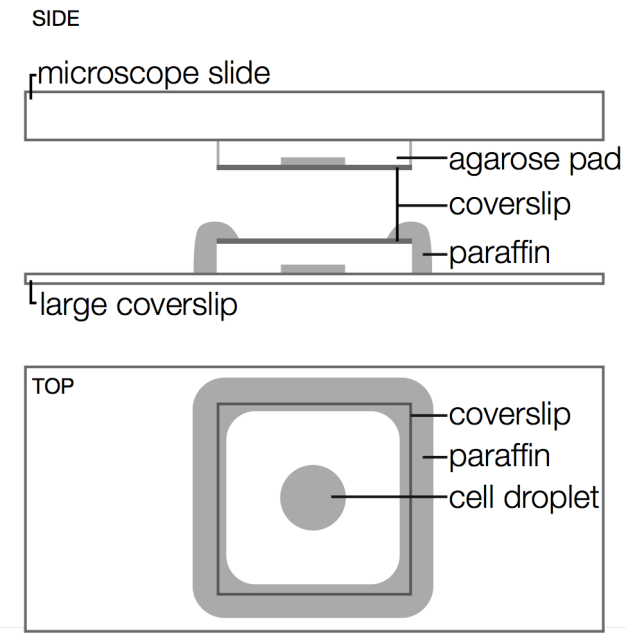

Figure 2.10: Microscopy sample geometry
Top: unsealed sample on a microscope slide.
Middle and Bottom: sample sealed with paraffin wax.
Side views are oriented for an inverted microscope setup.

of PA-mCherry (405 nm: 0.006–0.2 µW/µm^2 for 50–100 ms at a time, 561 nm: 0.2–0.3 mW/cm^2 for several minutes). One can also verify cell viability by imaging on a heated objective and watching for cell division, or by culturing cells from the sample after the microscopy experiment (Deich et al., 2004).

2.5.3 Drift

Even when the cells are stationary, the microscope stage may drift noticeably over time, either laterally or axially (i.e. in plane or out of focus). If the lateral drift is slow, it can be neglected for single-molecule tracking experiments, which measure relative positions between consecutive frames (steps), but in super-resolution PALM or STORM experiments, where the positions of molecules are recorded over a longer time, correcting for stage drift is essential. Fiducial markers on the sample, such as quantum dots or fluorescent beads, can be used in post-processing to register imaging frames onto a common frame of reference (Biteen and Moerner, 2010). Quantum dots are small compared to bacteria (<10 nm in diameter) and bright, but they also tend to blink frequently (Mahler et al., 2008), which can complicate the image alignment process. Fluorescent beads—microspheres stained with small molecule dyes—can also be used and are commercially available in several colors, and as multicolor compilations (e.g. TetraSpeck microspheres from Life Technologies; Biteen and Moerner, 2010). These beads are larger than quan-

tum dots (>100 nm in diameter), but because their signals come from many fluorophores, they are unlikely to blink. In some cases, however, we have found that the commercially available fluorescent beads shed dye molecules into the sample, which adsorb to the cell and agarose pad surfaces, or are even taken up by the cells, increasing the background fluorescence and confusing peak-selection algorithms.

While lateral drift can be corrected in post-processing as described above, in typical two-dimensional single-molecule experiments, it is more difficult to compensate for focus drift. Stage drift of this type can be corrected manually, with fine focus or a piezoelectric objective positioner, or automatically, by using a piezo stage controlled by a feedback loop. Several labs have taken the automatic focus control one step farther to do three-dimensional tracking of nanoparticles and fluorescently-labeled viruses in eukaryotic cells (Peters et al., 1998; Dupont and Lamb, 2011; Welsher and Yang, 2014).

2.5.4 Sources of background

Controlling background fluorescence is important in single-molecule imaging because the individual fluorophores have small signals that are easily overwhelmed. Autofluorescence due to flavins in the cell can be considerable in the yellow-green region of the spectrum (Benson et al., 1979; Yu et al., 2006). Using red fluorophores with appropriate emission filters can reduce the observed background. Our preferred label for imaging and tracking proteins in live *V. cholerae* is PAmCherry (absorption $\lambda_{\max}$ =564 nm, emission $\lambda_{\max}$ =595 nm); we excite this fluorescent protein with a 561-nm laser and filter out scattered laser light by approximately twelve orders of magnitude with a combination of a 580-nm long-pass dichroic mirror and a 580-nm long-pass filter (Fig. 2.7).

Additionally, since some percentage of photoactivatable or photoshiftable fluorescent proteins will be activated by ambient light at room temperature, we have found a "pre-bleaching" step helps to decrease PAmCherry background: the sample is exposed to the excitation laser (for 2–3 min at 0.2–0.3 µW/µm^2) to bleach pre-activated fluorophores, cellular autofluorescence, and other background sources before the first subset of fluorescent labels is intentionally photoactivated.

2.5.5 Balancing speed with precision

The precision of each single-molecule localization depends on the number of photons observed (Thompson et al., 2002). Since wide-field single-molecule fluorescence data are typically recorded on a pixelated electron multiplying charge-coupled device (EMCCD) detector, the duration of each observation depends on the camera frame rate. Lengthening the integration time (i.e. the image exposure time) will increase the duration of each observation and thus the detected photon count, but at a cost: the emission from moving molecules will be blurred over multiple detector pixels if the duration of an imaging

frame is too long, increasing the uncertainty in the position and losing information about molecular dynamics. For example, a cytoplasmic protein ($D \sim 7\,\mu m^2/s$) can diffuse the length of a *V. cholerae* cell in approximately 100 ms (Kumar et al., 2010), which is twice the duration of a typical imaging frame. The emission from this labeled protein will be spread out over the whole cell (500 pixels in our configuration) and thus be unresolvable. Increasing the excitation laser power can also increase the number of photons detected, but fluorophores bleach more rapidly in this case, and so the length of time a particular molecule can be observed is reduced. For single-molecule tracking experiments, higher laser powers therefore decrease observed trajectory lengths, severely curtailing the information content of an experiment (Michalet and Berglund, 2012).

We imaged TcpP-PAmCherry molecules in *V. cholerae* at three different integration times: 20 ms, 50 ms, and 100 ms. In the 100-ms frames, foci were brighter, but fast motions were lost. At an integration time of 20 ms per frame, we could detect faster diffusion, but since fewer photons were collected per frame, fluorophores also bleached faster because higher laser powers were required to observe the molecules above background. In these experiments, 50 ms frames were the best compromise between temporal and spatial resolution.

To overcome the limited time resolution of EMCCD detectors (maximum frame rate ~100 Hz), stroboscopic illumination can give rise to much shorter effective image integration times. Xie *et al.* adapted high-speed photography tricks for single-molecule imaging to track the very rapid motions of cytoplasmically soluble proteins in *E. coli* (Yu et al., 2006; Xie et al., 2008). By using very short (<0.5 ms) bursts of very intense (300 W/cm^2) excitation light coupled with long acquisition times, these experiments achieved single-molecule tracking with sub-millisecond time resolution. This stroboscopic microscopy method was applied to the single-molecule detection of gene expression products that diffused too fast to be localized with conventional illumination.

2.6 Analysis methods

2.6.1 Localization

Localization microscopy relies on post-processing analysis to create super-resolution reconstruction images. In each frame, fluorescence intensity maxima are identified, the intensity distributions are fit to a model function, typically a two-dimensional Gaussian function, and the position of each focus is taken to be the center of the model function with some localization uncertainty, for example the 95% confidence interval of the fit (Biteen et al., 2008). These position data can then be used to construct super-resolution maps of the fluorescently labeled molecules, to form single-molecule trajectories, and to calculate dynamic information such as the diffusion coefficient.

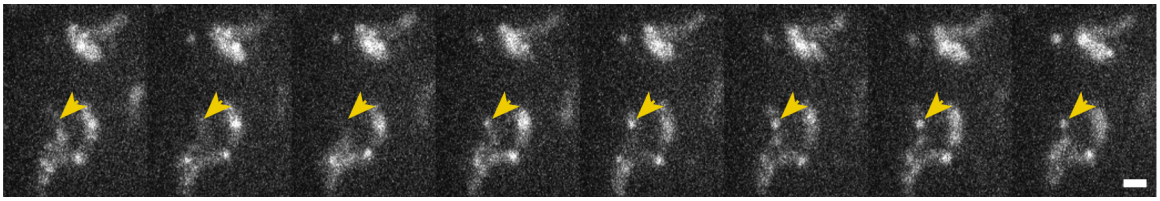

Figure 2.11: Time-lapse images of diffusing TcpP-PAmCherry molecules in live *V. cholerae* cells
The arrowhead highlights one molecule, which turns on in the fourth panel and moves inside the cell. Each panel is 50 ms. Scale bar is 1 µm.

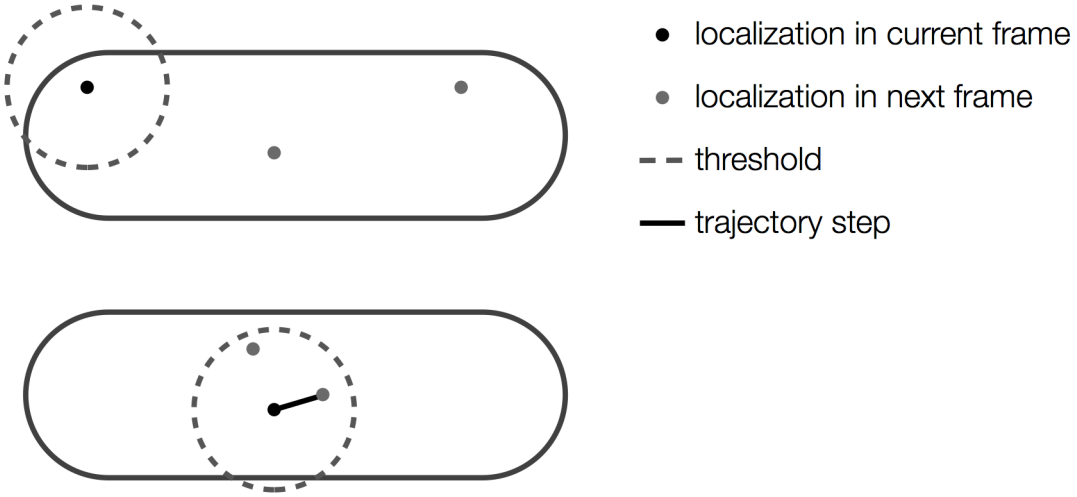

Figure 2.12: Implementing a step size threshold
Top: The molecule cannot be connected to either localization in the next frame because they are both beyond the set threshold.
Bottom: If more than one localization is within the threshold, the nearest is chosen.

2.6.2 Single-particle tracking

The common measure of Brownian motion is the diffusion coefficient, D. To determine D for a particular molecule, individual localizations are connected from frame to frame to form single-molecule trajectories (Fig. 2.11). The simplest method is a nearest-neighbor connection: two molecules located nearest to each other in consecutive frames are connected to form a trajectory. A threshold may be used to prevent nonsensical connections (Fig. 2.12). The main disadvantages of this method are the difficulties in choosing an appropriate threshold for maximum step size and the abrupt termination of tracks upon fluorophore blinking. Additionally, if the fluorophore density is too high, trajectories may cross, making track assignment untrustworthy (Fig. 2.8).

We chose our threshold of 300 nm per 50-ms frame (that is, consecutive localizations more than 300 nm apart were recorded as distinct molecules) for tracks of TcpP-PAmCherry in *V. cholerae* by running the tracking algorithm without a step size limit in

place and then examining the histogram of resultant step sizes. Most steps were smaller than 150 nm, though the distribution tail reached to ∼300 nm. We then ran the tracking algorithm a second time with the 300-nm threshold in place to filter out nonsensical cases, such as when a putative track might be formed between different molecules at opposite ends of the cell.

A number of algorithms have been developed to handle merging, splitting and crossing tracks, as well as blinking (Jaqaman et al., 2008; Jaqaman and Danuser, 2009; Shuang et al., 2013). Global fitting schemes have also been used, wherein the optimum connections are found for all localizations simultaneously, rather than sequentially (Jaqaman et al., 2008). For our studies of TcpP diffusion in *V. cholerae*, we used the simplest nearest-neighbor algorithm and relied on low fluorophore concentrations to prevent tracks from crossing. We did not allow blinking in trajectories because we were not confident enough that the putative steps taken during blinking frames were the same molecule. For single-frame steps, our step size limit was approximately the cell radius; with a single dark frame allowed for blinking molecules, this step size limit became a large fraction of the cell length.

Alternatively, one can skip particle-tracking altogether and use correlation analysis over time to extract diffusion coefficients. Two such methods are Particle Image Correlation Spectroscopy (PICS) and Spatio-Temporal Image Correlation Spectroscopy (STICS) (Hebert et al., 2005; Semrau and Schmidt, 2007; Di Rienzo et al., 2013). The former measures diffusion coefficients based on correlation of sequential particle localization maps, while the latter forgoes single-molecule localization and determines diffusion coefficients based on correlating the raw data of the imaging frames themselves. These two methods are sensitive for even low-signal or high fluorophore concentration data, though they do not provide maps of single-molecule trajectories, which are useful for determining whether the molecules have preferred motions in certain regions of the cell.

2.6.3 Mean squared displacement analysis

Whether successive localizations are connected as tracks, or motion is discerned from correlation, translating molecular displacements into a diffusion coefficient requires a model. There are several models for diffusion; the simplest is Brownian motion, in which the diffusion coefficient of a moving molecule is proportional to the slope of its mean squared displacement (MSD, $\langle r^2 \rangle$) curve, plotted as a function of time-lag, τ:

$$\left\langle r^2 \right\rangle = 2nD\tau \tag{2.1}$$

where n is the number of dimensions and D is the diffusion coefficient (Anderson et al., 1992). Other models exist for anomalous, directed and confined motions (Saxton, 2007).

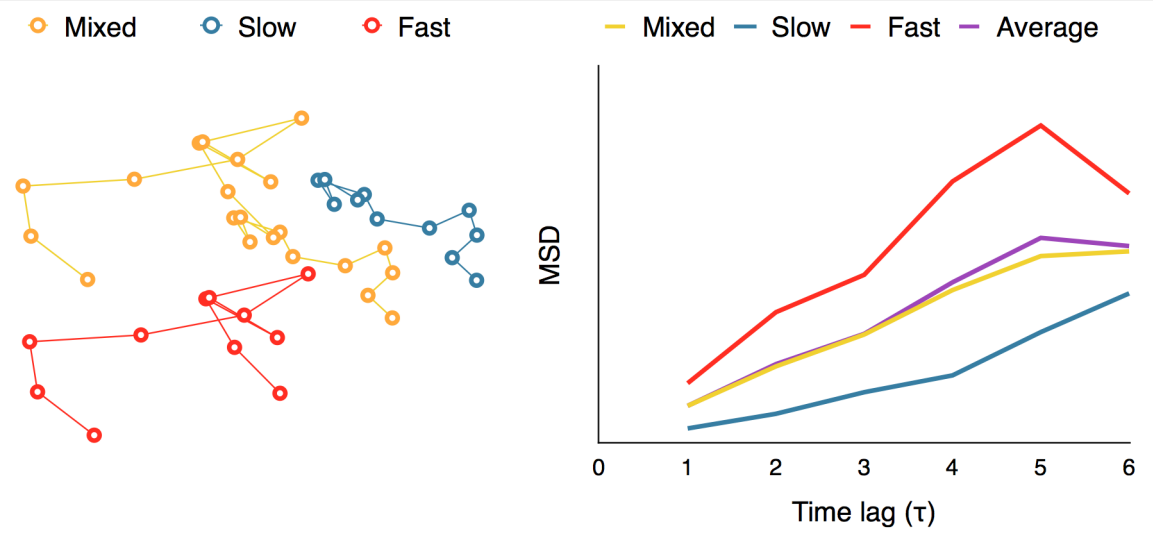

Figure 2.13: MSD averaging effect
A: Three simulated trajectories for molecules exhibiting fast (red) and slow (blue) diffusion, and a trajectory with a combination of the two (yellow).
B: MSD curves for the three trajectories in A, as well as the curve for the mean (purple) of the fast and slow MSDs.

Though it is tempting to categorize diffusive behavior based on the shape of single-trajectory MSD curves, it is important to remember that the stochastic nature of Brownian diffusion can make molecular motion appear confined or anomalous when it is in fact not (Robson et al., 2012).

Short trajectories, which are very common in live-cell microscopy, can be very noisy (Saxton, 1997). For MSD analysis, only points in the first 50-60% of time-lags are plotted because MSD values at larger values are averages drawn from few points and therefore have large errors (Qian et al., 1991; Saxton, 1997). This commonly-used threshold is somewhat arbitrary (Michalet, 2011). A statistically rigorous method for determining the optimum number of points to use when fitting MSD curves has been proposed instead (Michalet, 2011). Based on this method and our minimum trajectory length threshold of 10 frames, we chose to calculate diffusion coefficients using the first four points of the MSD curves.

The major drawback of mean squared displacement analysis is that it assumes each of the individual molecules exhibits homogeneous motion. For trajectories in which molecules change behavior, e.g. a protein slows down as it binds its target, simply extracting D from the slope of the MSD curves assigns diffusive roles that reflect some average behavior (Fig. 2.13).

2.6.4 Cumulative probability distribution analysis

An analytical tool that considers all steps independently of the trajectories they come from can include models for heterogeneous motion and overcome the limitations of MSD analysis described above. One such method is the cumulative probability distribution (CPD). For two-dimensional motion, the CPD describes the probability of a molecule staying within an area defined by a radius, r, given localization accuracy, σ, during a given time-lag, τ (Schütz et al., 1997). This model includes one exponential term per population of molecules. Analyzing the CPD by fitting it to a multi-component model (Eq. 2.2) provides a framework for considering heterogeneous mixtures of molecular populations.

$$P(r^2,\tau) \ = 1-\alpha\cdot\exp\left(\frac{-r^2}{\langle r_\alpha^2\rangle+4\sigma^2}\right)-\beta\cdot\exp\left(\frac{-r^2}{\left\langle r_\beta^2\right\rangle+4\sigma^2}\right)-\ldots \tag{2.2}$$

Fitting this model to the step size CPD gives the fraction of molecules in each population (α, β, etc., where $\alpha+\beta+\ldots=1$), as well as the MSD value for each population at each time-lag ($\langle r_\alpha^2(\tau)\rangle$, $\left\langle r_\beta^2(\tau)\right\rangle$, etc.). The MSD values are then plotted as in single-molecule MSD analysis (Eq. 2.1), and the diffusion coefficients for each population are calculated from the slopes of these curves using the first four points, as described in Section 2.6.3. For both single-molecule MSD and CPD analysis of TcpP diffusion in *V. cholerae*, we included only trajectories with at least 10 frames, removing very noisy datasets from consideration.

The major trade-off for this ability to model heterogeneity, though, is that this stepwise analysis does not permit the classification of individual proteins into these resulting populations. For example, in our studies of TcpP-PAmCherry diffusion in *V. cholerae*, we found a three-term model described the data the best, with one "fast" population, one "slow" population, and one population of molecules that were immobile ($\left\langle r_\gamma^2\right\rangle = 0\,\mu m^2/s$) within our resolution (∼40 nm). Simpler models, with only one or two terms, did not describe the data well (Fig. 2.14). The best-fit model allowed us to determine that, at any given time, 22% of TcpP-PAmCherry molecules in a Δ*tcpP* TcpP-PAmCherry strain are in the immobile population, but the aggregation of all trajectory data prevents us from determining which specific molecules are immobilized, or to which trajectories they belong. On the other hand, single-molecule MSD analysis could not give us such clear population information.

2.7 Curvature challenges

Bacteria are small enough (typically 0.5–1 µm in diameter; Koch, 1996) to fit entirely within the focal depth of a light microscope (0.703 µm; Qiu et al. 2012). But even though they are very thin, they are not two-dimensional; the actual locations and motions inside

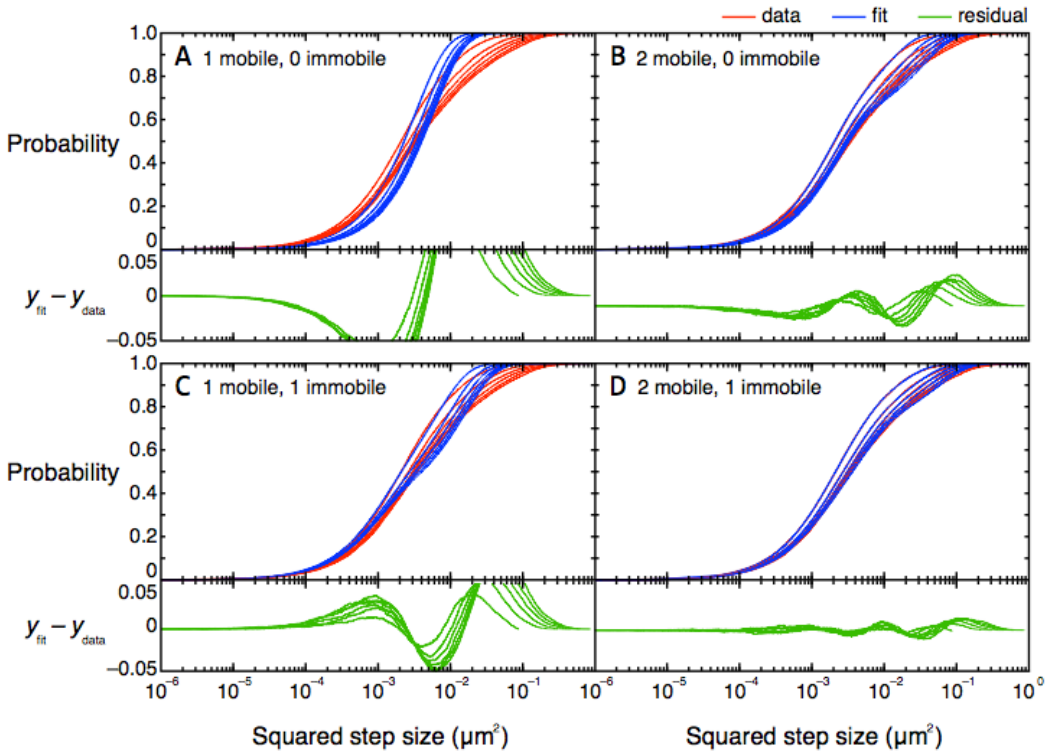

Figure 2.14: CPD curves and alternative models for TcpP-PAmCherry diffusion
A three-term model, with two mobile populations and one immobile population, described the WT* CPD data best (panel D). Models with fewer terms (panels A–C) fit poorly, as seen in the plots of their residuals (green curves). Experimental data are shown in red and best fits for each of the given models are shown in blue. Curves for the first six time lags (50–200 ms) are plotted left to right. Residuals (i.e. the differences between each of the models and the experimental results) are shown in green.

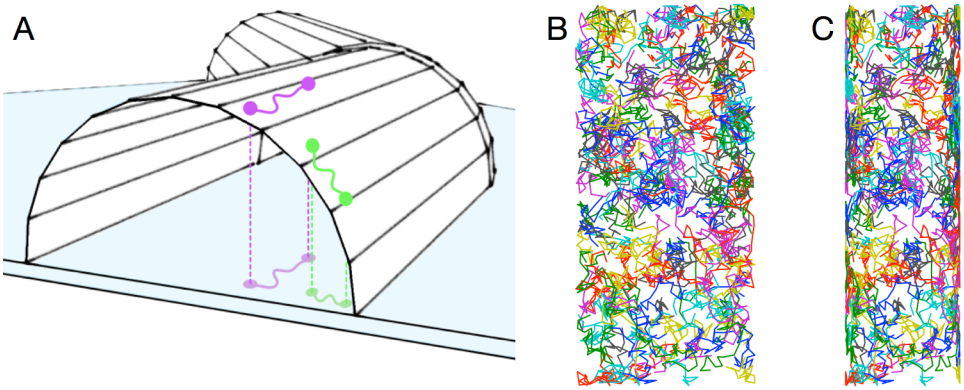

Figure 2.15: Effect of cell curvature on membrane-bound trajectories
A: The purple and green paths are the same length on the surface of the cell, but when projected onto the imaging plane (represented by the blue surface), the green path appears shorter than the purple path due to the curvature of the cell membrane.
B: 200 two-dimensional random walks on the surface of a cylinder.
C: The projection of the walks in B onto a simulated two-dimensional detector.

a bacterium may appear distorted in two-dimensional images. For example, a fluorescent molecule moving perpendicular to the focal plane through the cytoplasm will appear stationary in a movie of the cell. Even the two-dimensional motion of a membrane-bound protein is distorted when projected onto the imaging plane (Fig. 2.15). Steps taken parallel to the short axis of the cell, especially near the edge, appear shorter than those along the long axis of the cell, leading to underestimations of the speed—and therefore underestimates of the diffusion coefficient—of the protein (Deich et al., 2004; van den Wildenberg et al., 2011).

We modeled a random walk on the upper surface of a cylinder with the same dimensions as *V. cholerae* to examine the effect of membrane curvature on our tracking results. In our model, the molecule moved on a two-dimensional surface, which was then transformed into three dimensions to simulate motion on a cylindrical surface and projected onto a second two-dimensional surface to simulate detection in two dimensions by an EMCCD detector (Fig. 2.15). In this way, we could compare the actual two-dimensional motion with the detected motion. The two-dimensional projection clearly shows distortion due to the curvature of the membrane (Fig. 2.15). To quantify this distortion, we simulated 200 molecules taking steps chosen from a distribution (corresponding to a diffusion coefficient of 0.066 $\mu m^2/s$). The best-fit CPD model for the projection of the walk had two terms: a large (∼95%) mobile population (with an apparent diffusion coefficient of 0.057 $\mu m^2/s$) and a small (4–6%) immobile population. The immobile population can be attributed entirely to curvature artifacts. In the experimental data for TcpP-PAmCherry diffusion in *V. cholerae*, an immobile population was present in three cell strains, but it was significantly larger (10–20%) than could be attributed to the curvature artifact alone.

We concluded that there does in fact exist a nontrivial population of TcpP molecules that are immobile within our 40-nm resolution.

There are several ways to account for the effects of imaging inherently three-dimensional motion on a two-dimensional plane. For example, one can explicitly take the shape of the cell into account and use simulations to find a conversion factor back to the actual diffusion coefficient (Deich et al., 2004). Alternatively, one can manipulate the point spread function to obtain information about the position of the molecule in the z-axis (i.e. perpendicular to the imaging plane). In one such setup, a cylindrical lens is inserted into the emitted light path, distorting the symmetric point spread function (PSF) of a dipole's emission into an elliptical PSF whose aspect ratio depends on the z-position. By calibrating the degree of this astigmatism at known heights, the z-position can be extracted for single molecules in unknown positions (Huang et al., 2008; Biteen et al., 2012). Another example of this emission-beam engineering is the creation of a double-helix point spread function. By manipulating the emission beam in Fourier space, this method also yields a PSF that changes with z-position, and z information can be obtained from a calibration. In the latter case, the emitter appears as two points, rather than one, and the angle (relative to a specified axis) between the two points changes as a function of z-position (Pavani et al., 2009; Lee et al., 2012). Additional approaches to three-dimensional imaging are reviewed in Fischer et al., 2011.

2.8 Conclusions

There are many obstacles to consider when designing a single-molecule experiment in live cells, but they are not insurmountable. Indeed, several groups have used super-resolution localization microscopy and single-molecule tracking to study molecular-scale interactions in live bacteria, including FtsZ structure and MreB diffusion in *C. crescentus* (Kim et al., 2006; Biteen, 2012), gene transcription in *E. coli* (Yu et al., 2006; Elf et al., 2007), nucleoid associated proteins in both *C. crescentus* and *E. coli* (Lee et al., 2011; Wang et al., 2011), and facilitated diffusion of the *lac* repressor in *E. coli* (Hammar et al., 2012).

In the next chapter, we will use super-resolution localization microscopy and single-protein tracking to investigate transcription activator protein dynamics in the human pathogen *V. cholerae*.

Chapter 3

Single-molecule tracking in live *Vibrio cholerae* reveals that ToxR recruits the membrane-bound transcription activator TcpP to the *toxT* promoter[1]

3.1 Introduction

Cholera is an ancient waterborne disease that still kills in modern times, as evidenced by the 2010 outbreak in Haiti (Cravioto et al., 2011). *V. cholerae* bacteria cause this epidemic disease by colonizing the human gut, where they produce cholera toxin (Mekalanos, 2011). Although rehydration therapy and antibiotics can treat the symptoms and stop the bacteria from causing further harm to a patient (Matson et al., 2007), cholera remains a threat to human health in the developing world. Understanding the mechanisms by which *V. cholerae* produces its potent toxin will provide avenues for developing novel therapeutic approaches.

The cytoplasmic ToxT protein activates transcription of cholera toxin and other associated virulence factors (Beck et al., 2004), but ToxT expression is regulated by an unusual membrane-associated mechanism: the bitopic proteins TcpP and ToxR cooperate to activate toxT transcription while remaining restricted to the inner membrane (Fig. 3.1A; Matson et al. 2007). TcpP is the direct *toxT* transcription activator, while ToxR plays an accessory role, enhancing DNA-binding or transcription activation by TcpP (Häse and Mekalanos, 1998; Krukonis et al., 2000; Matson et al., 2007). The co-localization of TcpP and ToxR, and their interactions with other components of the transcription complex,

[1]In collaboration with Jyl S. Matson and Victor J. DiRita. Bacterial strain construction and immunodetection were carried out by Jyl Matson. qRT-PCR analysis was carried out by Aaron Konopko and Jyl Matson.

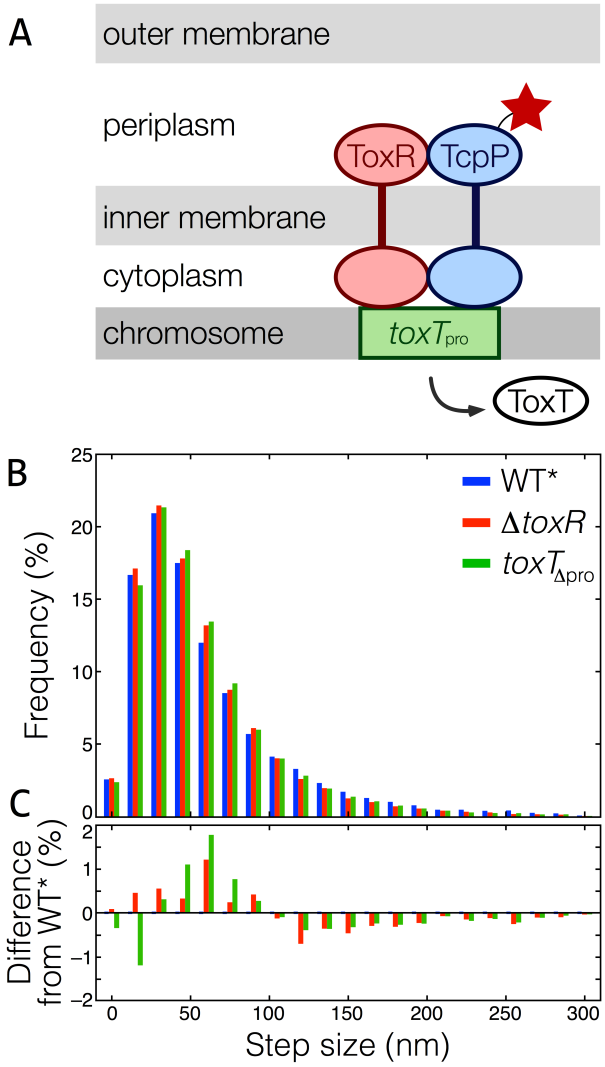

Figure 3.1: ToxR regulon and TcpP step sizes
A: Schematic of the ToxR regulon. ToxR and TcpP bind the *toxT* promoter ($toxT_{\text{pro}}$) and activate transcription of ToxT. In this study, TcpP is labeled with the fluorescent protein TcpP-PAmCherry in the periplasm (star).
B: Histogram of sizes of steps taken between consecutive 50-ms imaging frames in the three mutants: WT* (blue), Δ*toxR* (red), and $toxT_{\Delta\text{pro}}$ (green). The histogram data are aggregated across multiple experiments and normalized by the total number of steps taken. The number of cells, trajectories, and steps measured are listed in Table 3.4.
C: Differences between the histograms of the knockout strains (Δ*toxR*, red, and $toxT_{\Delta\text{pro}}$, green) and the WT* strain.

have not been observed directly in cells; the biophysical details of these protein–protein and protein–DNA interactions therefore remain unclear.

TcpP and other proteins of interest are too small to see using traditional light microscopy, and, due to the short timescale of protein–protein interactions, they are also too dynamic for electron microscopy. Single-molecule fluorescence (SMF) microscopy bridges this gap between conventional microscopy and electron microscopy, and reveals

dynamic interactions occurring on size scales below the ∼300 nm diffraction limit of visible light (Qian et al., 1991; Thompson et al., 2002) The high sensitivity of SMF imaging enables detection of isolated fluorescent labels using traditional wide-field microscopy. The center of each emission spot indicates the position of the individual molecule with 10–40 nm resolution, even for proteins in living cells, where fluorescent protein fusions provide highly specific labels that can be detected, mapped and tracked non-invasively (Xia et al., 2013). These powerful methods have been applied to bacteria (Xie et al., 2008; Biteen and Moerner, 2010; Wang et al., 2011; Bakshi et al., 2012), and are now being extended beyond model bacterial systems to infectious microbes (Berk et al., 2012).

Based on super-resolution reconstructions with Photoactivated Localization Microscopy (PALM; Betzig et al., 2006; Rust et al., 2006; Hess et al., 2006) and quantitative analysis of single-molecule trajectories (Qian et al., 1991; Schütz et al., 1997; Lommerse et al., 2004; Kim et al., 2006; Jaqaman et al., 2008), we uncover the molecular-scale interactions and dynamics of TcpP in live *V. cholerae*. In particular, we find that the heterogeneous movements of TcpP are categorized into three populations: one fast, one slow, and one immobile. By comparing TcpP diffusion with and without the possibility of interacting with ToxR and the *toxT* promoter, we find that TcpP moves faster in the presence of both binding partners than it does in cells lacking either one of those factors ($\Delta toxR$ and $toxT_{\Delta\text{pro}}$, respectively), which implies that (1) binding at the *toxT* promoter mediates the interaction between ToxR and TcpP, and (2) ToxR increases TcpP mobility. Our findings support a mechanism in which ToxR recruits TcpP to the *toxT* promoter for *toxT* transcription activation.

3.2 Methods

3.2.1 Bacterial strains and plasmid construction

The *V. cholerae* classical strain O395 was used throughout this study. The *Escherichia coli* strain JM101 was used for cloning. The *tcpP-PAmCherry* chimeric gene was constructed as follows: *tcpP* (lacking the stop codon) was amplified from O395 chromosomal DNA using Expand Hi-Fidelity polymerase. Photoactivatable (PA)-mCherry was amplified from pPAmCherry N1 (Clontech; Subach et al., 2009b). A secondary SOEing PCR using the *tcpP* and PAmCherry PCR products as template DNA was used to generate the fusion gene. The resulting PCR product was digested with EcoRI and XbaI and ligated into similarly digested pBAD18-Kan (Guzman et al., 1995). The resulting plasmid was confirmed by sequencing and transformed into *V. cholerae* strains by electroporation. The TcpP-PAmCherry fusion protein was expressed in the following previously described strains for visualization and localization studies: O395 $\Delta tcpP$ (RY1; Yu and DiRita,

Table 3.1: Summary of strains used for imaging
Cells strains and abbreviations used. TcpP-PAmCherry was expressed in all three strains on plasmid pBAD18-Kan (Guzman et al., 1995).

	Strain	Reference
WT*	O395 Δ*tcpP* (RY1)	Yu and DiRita, 1999
Δ*toxR*	O395 Δ*toxR* Δ*tcpP* (EK459)	Krukonis et al., 2000
$toxT_{\Delta\text{pro}}$	O395 $toxT_{\Delta\text{pro}}$ Δ*tcpP* (EK1647)	Goss et al., 2010

1999), O395 Δ*tcpP* Δ*toxR* (EK459; Krukonis et al., 2000), and O395 Δ*tcpP* $toxT_{\Delta\text{pro}}$ (EK1647; Goss et al., 2010; Table 3.1).

3.2.2 Protein electrophoresis and immunodetection

Overnight cultures of *V. cholerae* were subcultured 1:100 in pH 6.5 LB and grown for 4 hours at 30 °C. Arabinose was added to the culture medium at the time of subculture for strains containing pBAD18-Kan. 1 mL of mid-log culture was pelleted by centrifugation and resuspended in 1× sample buffer. Proteins were separated by SDS-polyacrylamide gel electrophoresis (SDS-PAGE) using 15% (wt/vol) polyacrylamide gels. Samples were boiled for 5 minutes before loading on the gels and loading volumes were adjusted to normalize for culture OD_{600}. Proteins were then transferred to nitrocellulose membranes and probed with rabbit anti-TcpP antibodies (generated by Rockland) or rabbit anti-TcpA antibody (generously supplied by Dr. Ronald Taylor). Blots were probed with goat anti-rabbit AP-conjugated secondary antibody (Zymed) followed by visualization using NBT-BCIP (nitroblue tetrazolium and 5-bromo-4-chloro-3-indolylphosphate; Roche).

3.2.3 qRT-PCR analysis

Triplicate cultures of O395, O395 Δ*tcpP*, and WT* were grown and induced as above. Equivalent numbers of cells from each sample were harvested and RNA was extracted using TRIzol reagent (Life Technologies). RNA samples for qRT-PCR were DNase treated, run on an agarose gel to check quality, and quantified by measuring the OD_{260}. 2.5 µg of each sample were treated with Moloney murine leukemia virus (M-MLV) reverse transcriptase (Invitrogen) according to the manufacturer's specifications. For detection of transcripts, primers amplifying a 200-bp region in the center of the mRNA were used with SYBR Green Master Mix (Stratagene) on a Stratagene MX3000P thermocycler. Primers were designed using the OligoPerfect tool (Invitrogen). Each test was performed in triplicate at least three times, and fold change in expression was calculated using the ΔΔCT method with *recA* transcript levels as the reference.

3.2.4 Cell growth and sample preparation for microscopy

Bacterial cultures were grown in LB medium at 37 °C with shaking (180 rpm), then transferred to M9 minimal medium with 0.4% glycerol and an amino acid supplement (asparagine, arginine, glutamic acid and serine, 25 mM final concentration) and grown to turbidity at 30 °C. Kanamycin (50 µg/mL final concentration) was used to select for the plasmid. Arabinose (0.1% final concentration) was added to the cultures to induce expression of the fusion protein and the cultures were incubated for an additional 4 hours. A 1-mL aliquot of culture was centrifuged for 30 s at 30,000×g to pellet the cells. The pellet was washed in 1 mL warm M9, and centrifuged a second time. The supernatant was then removed, and the cell pellet was resuspended in a minimum of residual liquid (<100 µL). A 2.0-µL droplet of concentrated cells was then placed onto an agarose pad (2% agarose dissolved in M9, spread on a microscope slide) and covered with a coverslip.

3.2.5 Super-resolution microscopy

Samples were imaged at room temperature using wide-field epifluorescence microscopy on an Olympus IX71 inverted microscope with a 100× 1.40 NA oil immersion objective (Zeiss Immersol 518F immersion oil), a Semprex micrometer stage, a PIFOC piezo element, and appropriate excitation, emission, and appropriate dichroic filters (Semrock LL01-561, Semrock BLP01-561, Semrock Di01-R561). A Photometrics Evolve EMCCD camera with $> 90\%$ quantum efficiency was used to capture the images at 20 frames per second. Each pixel of the detector corresponds to a 49 × 49 nm area of the sample. Fluorescence of PAmCherry in the cells was activated using a 405-nm laser (Coherent Cube 405-100), co-aligned with the 561-nm fluorescence excitation laser (Coherent Sapphire 560-50). The lasers were both operated at low power densities ($0.006 - 0.2\,\mu W/\mu m^2$ and $0.2 - 0.3\,\mu W/\mu m^2$, respectively). The excitation and activation pathways were coupled by a dichroic mirror (Semrock Di01-R405), and both laser beams were circularly polarized (Tower Optical AO15Z 1/4 556, Tower Optical AO15Z 1/4 408). To prevent higher-order excitation during photo-activation, a pair of Uniblitz shutters controlled the laser beams such that samples were exposed to only one laser at a time. During imaging, the cells were given a 70-ms dose of 405-nm light every 90 s. Acquisitions lasted 5–7 minutes each.

Single fluorophores were localized using custom MATLAB code (Biteen et al., 2008) that detects diffraction-limited local intensity maxima above a specified background threshold and fits their point-spread functions (PSFs) to symmetric two-dimensional Gaussian functions (Thompson et al., 2002). The output of this code includes positions in x and y, and the 95% confidence intervals for these positions, which were used to estimate the localization error. The super-resolution reconstruction images in Fig. 3.2A, C and E show single molecule localizations blurred to these confidence intervals.

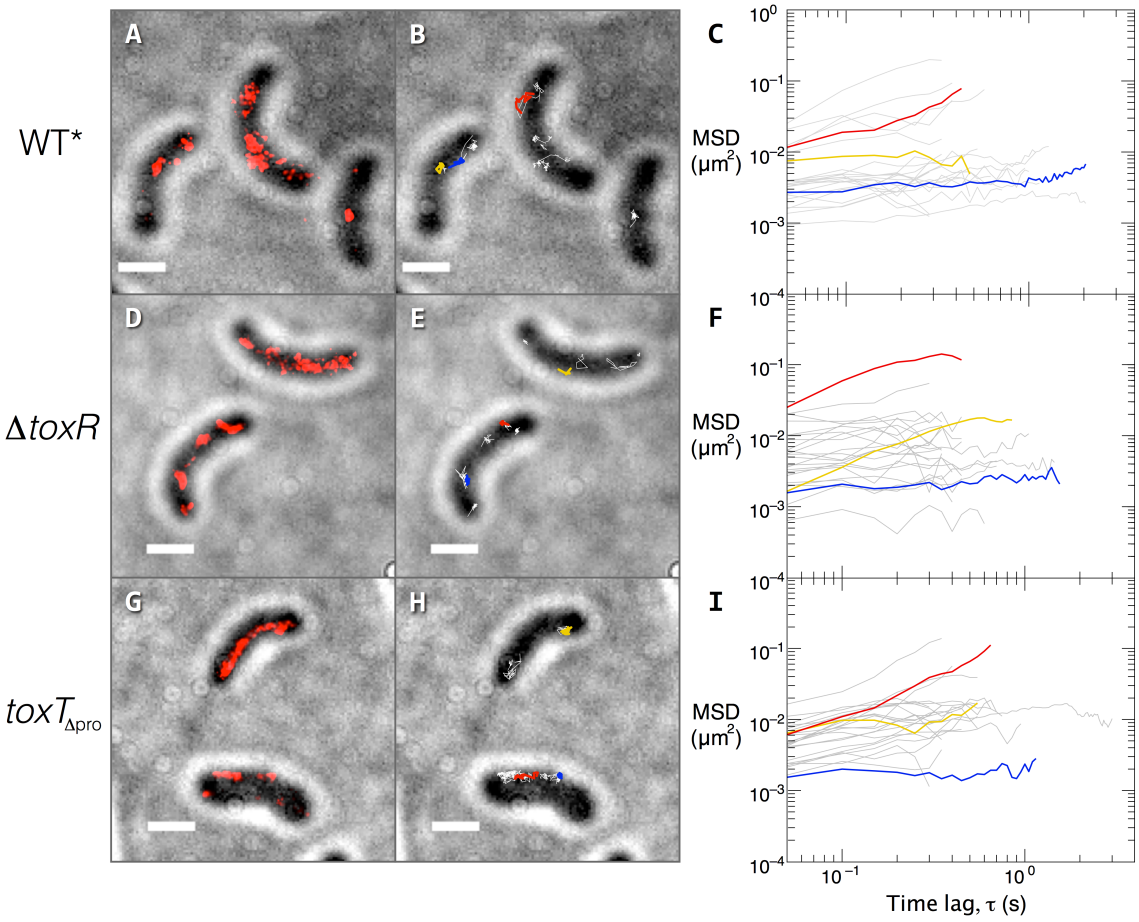

Figure 3.2: PALM, tracks and MSDs

A, D, and G: Super-resolution reconstructions (red) from 20–40 s of image capture overlaid on reverse-contrast bright-field images of WT*, Δ*toxR*, and $toxT_{\Delta\text{pro}}$ *V. cholerae* cells, respectively. Only molecules localized with resolutions better than 40 nm are included.
B, E, and H: Single-molecule trajectories from the same movie segments shown in A, D, and G, respectively (random colors) overlaid on the same reverse-contrast bright-field cell images. Only trajectories lasting at least 0.50 s (10 frames) are shown. Three trajectories are highlighted (red, yellow, blue) to show the variety of motion observed. Additional trajectories are plotted in white.
C, F, I: MSD curves for single TcpP-PAmCherry molecules in (C) WT* (F) Δ*toxR* and (I) $toxT_{\Delta\text{pro}}$ strains. The three colored curves (red, yellow and blue) in each plot correspond to the like-colored trajectories in B, D, and F, respectively. Gray curves correspond to the additional white trajectories. Plots show τ values up to two-thirds of the maximum τ. Scale bars: 1 μm.

Single-molecule trajectories were determined using custom MATLAB code employing a nearest-neighbor algorithm: molecules localized in consecutive frames within six pixels (294 nm) of each other were considered members of the same track. Only trajectories with at least 10 frames were used for further analysis.

3.2.6 Mean-squared displacements

For two-dimensional Brownian motion, the diffusion coefficient, D, of a molecule is proportional to the slope of its mean-squared displacement (MSD) curve:

$$\left\langle r^2 \right\rangle = 4D\tau \tag{3.1}$$

where τ is the time lag, i.e. the amount of time in which the displacements occur (Qian et al., 1991).

3.2.7 Cumulative probability distributions

For homogeneous two-dimensional Brownian motion, the probability of a molecule remaining within a circle of radius, r, in a given time lag, τ, is given by:

$$P\left(r^2, \tau\right) = 1 - \exp\left(\frac{-r^2}{\langle r^2 \rangle + \sigma^2}\right) \tag{3.2}$$

where σ is the localization accuracy (Sonnleitner et al., 1999; Lommerse et al., 2004; van den Wildenberg et al., 2011). Additional populations are incorporated by including additional exponential terms in the expression. We used a three-term model: two mobile terms with weights α and β, and one immobile term (i.e. $\langle r^2 \rangle = 0$) with weight $\gamma = 1 - (\alpha + \beta)$:

$$P\left(r^2, \tau\right) = 1 - \alpha \cdot \exp\left(\frac{-r^2}{\langle r_\alpha^2 \rangle + \sigma^2}\right) - \beta \cdot \exp\left(\frac{-r^2}{\left\langle r_\beta^2 \right\rangle + \sigma^2}\right) - \gamma \cdot \exp\left(\frac{-r^2}{\sigma^2}\right) \tag{3.3}$$

The errors in the fractions and mean-squared displacements obtained from CPD analysis were calculated by fitting the experimental CPD curves 100 times and finding the means and standard deviations of each parameter (i.e. α, β, $\langle r_\alpha^2 \rangle$ and $\left\langle r_\beta^2 \right\rangle$).

3.2.8 Monte Carlo simulations

Simulations were performed in MATLAB. The cell membrane was modeled as the surface of a half-cylinder 2 µm long and 0.6 µm in diameter, to match the dimensions of a *V. cholerae* bacterium (Kay et al., 1994). Each of 200 molecules was allowed to diffuse on this surface for 20 frames, starting at a random position. Steps were chosen from a

Figure 3.3: Western blot of *toxT*-regulated toxin coregulated pilus protein TcpA
Lane 1: wild type O395 *V. cholerae*
Lane 2: Δ*tcpP*
Lane 3: Δ*tcpP* TcpP-mCherry
Lane 4: Δ*tcpP* TcpP-PAmCherry
In a Δ*tcpP* strain, transcription of *toxT* does not occur, and the *toxT*-regulated pilus protein TcpA is not produced (lane 2). Insertion of a plasmid containing TcpP-mCherry or TcpP-PAmCherry restores transcription of *toxT* and *tcpA* under inducing conditions (lanes 3 and 4).

normal distribution corresponding to a diffusion coefficient of 0.066 $\mu m^2/s$. The three-dimensional molecular positions at each frame were then projected onto a plane parallel to the cell. CPD analysis of these projected steps revealed an apparent immobile population of 2–6%.

3.3 Results

3.3.1 *In vitro* characterization confirms TcpP-PAmCherry fusion activity

We monitored the motion of a TcpP-PAmCherry fusion expressed from an arabinose-inducible promoter in Δ*tcpP V. cholerae*. Immunoblotting with anti-TcpP serum demonstrated that the TcpP-PAmCherry fusion is intact and stable (Fig. 2.4). Real-time reverse-transcription PCR (qRT-PCR) revealed that Δ*tcpP* TcpP-PAmCherry cells induced under the same conditions used for the microscopy experiments (Section 3.2.4) activate *toxT* transcription at approximately the same level as wild-type cells (0.8×). The TcpP-PAmCherry protein fusion also complements the Δ*tcpP* strain for expression of the ToxT-controlled toxin-coregulated pilus protein TcpA (Fig. 3.3). Taken together, these results confirm that TcpP-PAmCherry is functional in *V. cholerae*, and that our labeled system is comparable to the wild type system, which has a fully functional ToxR regulon. We refer to this cell strain as WT* in subsequent discussion.

3.3.2 Live-cell single-molecule imaging reveals TcpP positions and trajectories

WT* *V. cholerae* cells grown in virulence-inducing conditions in minimal media with arabinose were imaged using 405-nm light for photo-activation and 561-nm light for ex-

citation. The TcpP-PAmCherry was initially invisible, but after photo-activation, one to three copies of TcpP-PAmCherry were visible at a time, and the emission from these molecules was recorded until they photobleached. Through repeated cycles of photo-activation and imaging over 5–7 minutes, 10–50 TcpP-PAmCherry molecules were detected in each cell. We determined the position of each TcpP molecule in each 50-ms imaging frame with sub-pixel resolution. PALM super-resolution images of the TcpP positions in the cells were reconstructed from these localizations (Fig. 3.2A). Furthermore, single-molecule trajectories were created by connecting TcpP-PAmCherry localizations within 294 nm in consecutive frames. Most single-frame steps were smaller than 100 nm (Fig. 3.1B). Only trajectories lasting at least 0.50 s (ten consecutive imaging frames) were included in further analysis. The PALM images and tracks of three representative cells in Fig. 3.2A and B, respectively, indicate that TcpP is dynamic and explores the *V. cholerae* membrane.

3.3.3 Single-molecule trajectory analysis measures dynamics

Fig. 3.2A and B qualitatively depict TcpP motion in *V. cholerae*. To understand TcpP dynamics and interactions, we compare TcpP motion across three *V. cholerae* strains, but to make meaningful comparisons a more quantitative analysis of the trajectories is required. Analysis of the mean squared displacement (MSD) of single-molecule tracks provides diffusion coefficients for the individual molecules observed (Fig. 3.2C, F, I). However, Eq. 3.1 can determine D from MSD only in the case of homogeneous motion, which precludes a straightforward interpretation in the case of a molecule that shifts from one mode of motion to another during a single trajectory due to encounters with obstacles or binding partners. In the present study, if TcpP scans DNA freely at first and then becomes immobile when it binds to the *toxT* promoter, a single-molecule MSD curve would not separate the two behaviors and would reflect some average behavior instead (Fig. 2.13; Michalet, 2011). To overcome this limitation of MSD and quantify distinct TcpP dynamics, we considered the cumulative probability distribution (CPD) of all TcpP steps rather than separate single-molecule tracks (Sonnleitner et al., 1999; Lommerse et al., 2004; van den Wildenberg et al., 2011). Here, we fit distributions of squared step sizes at each time lag, τ, to a model distribution (Eq. 3.3) that explicitly allows for heterogeneity in order to categorize the collective motion of all TcpP molecules into multiple populations, each with a distinct diffusion coefficient and relative contribution to the whole.

The CPDs for time lags of $\tau = 50$, 100, 150 and 200 ms are plotted in Fig. 3.4A in red, blue, green and black, respectively. These data are well described (fit shown with dashed grey curves) by a model (Eq. 3.3) with three different populations of TcpP motion: a fast population, a slow population, and a population that is immobile within our 40-nm

Table 3.2: Summary of cumulative probability distribution results
Fractional contributions for the "fast," "slow," and "immobile" terms (α, β, and γ, respectively). Errors are ±1 standard deviation.

Strain	α (%)		β (%)		γ (%)	
WT*	29.4	±0.6	48.4	±0.5	22.2	±0.8
Δ*toxR*	21.2	±0.7	53.7	±0.6	25.0	±0.9
$toxT_{\Delta pro}$	26.8	±0.6	59.0	±0.4	14.1	±0.7

Table 3.3: Summary of diffusion coefficients determined from cumulative probability distribution
Diffusion coefficients for the mobile terms (D_α and D_β). Errors are ±1 standard deviation.

Strain	$D_\alpha(\times 10^{-3}\,\mu m^2/s)$		$D_\beta(\times 10^{-3}\,\mu m^2/s)$	
WT*	35.1	±2.1	1.6	±0.4
Δ*toxR*	23.4	±3.1	0.8	±0.3
$toxT_{\Delta pro}$	18.8	±1.4	1.0	±0.3

resolution. Models with fewer terms fit poorly (Fig. 2.14). According to Eq. 3.3, the two mobile populations are each described by an MSD value at each time lag. Diffusion coefficients for each population can be determined from the slopes of their respective MSD curves (Fig. 3.4B; Eq. 3.1). The CPD and MSD curves in Fig. 3.4 reveal that 29.5% of the TcpP molecules in WT* contribute to a fast mobile population with $D_\alpha = 35 \times 10^{-3}\,\mu m^2/s$, almost half (48.4%) of the TcpP molecules diffuse an order of magnitude more slowly ($D_\beta = 1.6 \times 10^{-3}\,\mu m^2/s$), and the remaining 22.2% of the TcpP molecules are immobile within our <40 nm resolution. Importantly, the relatively small diffusion coefficients for both the fast and slow populations are too small to be attributable to the ∼2–5 $\mu m^2/s$ diffusion coefficients expected for freely diffusing protein in the periplasm (Sochacki et al., 2011). Fractional contributions of each term and diffusion coefficients for the mobile populations are presented in Tables 3.2 and 3.3, respectively, and compared in Fig. 3.5.

3.3.4 TcpP diffuses faster in the presence of ToxR and the *toxT* promoter

To investigate the impact of protein–protein and protein–DNA interactions on TcpP diffusion, we compared the TcpP-PAmCherry motion in WT* *V. cholerae* to TcpP-PAmCherry motion in strains lacking either the ToxR binding partner (Δ*toxR*) or the *toxT* promoter region ($toxT_{\Delta\text{pro}}$; position −112 to +1, relative to the start of *toxT* transcription; Krukonis et al. 2000; Goss et al. 2010; Table 3.1). The PALM images and tracks from live-cell imaging are shown in Fig. 3.2D and E for two representative Δ*toxR* cells, and in Fig. 3.2G and H for two representative $toxT_{\Delta\text{pro}}$ cells.

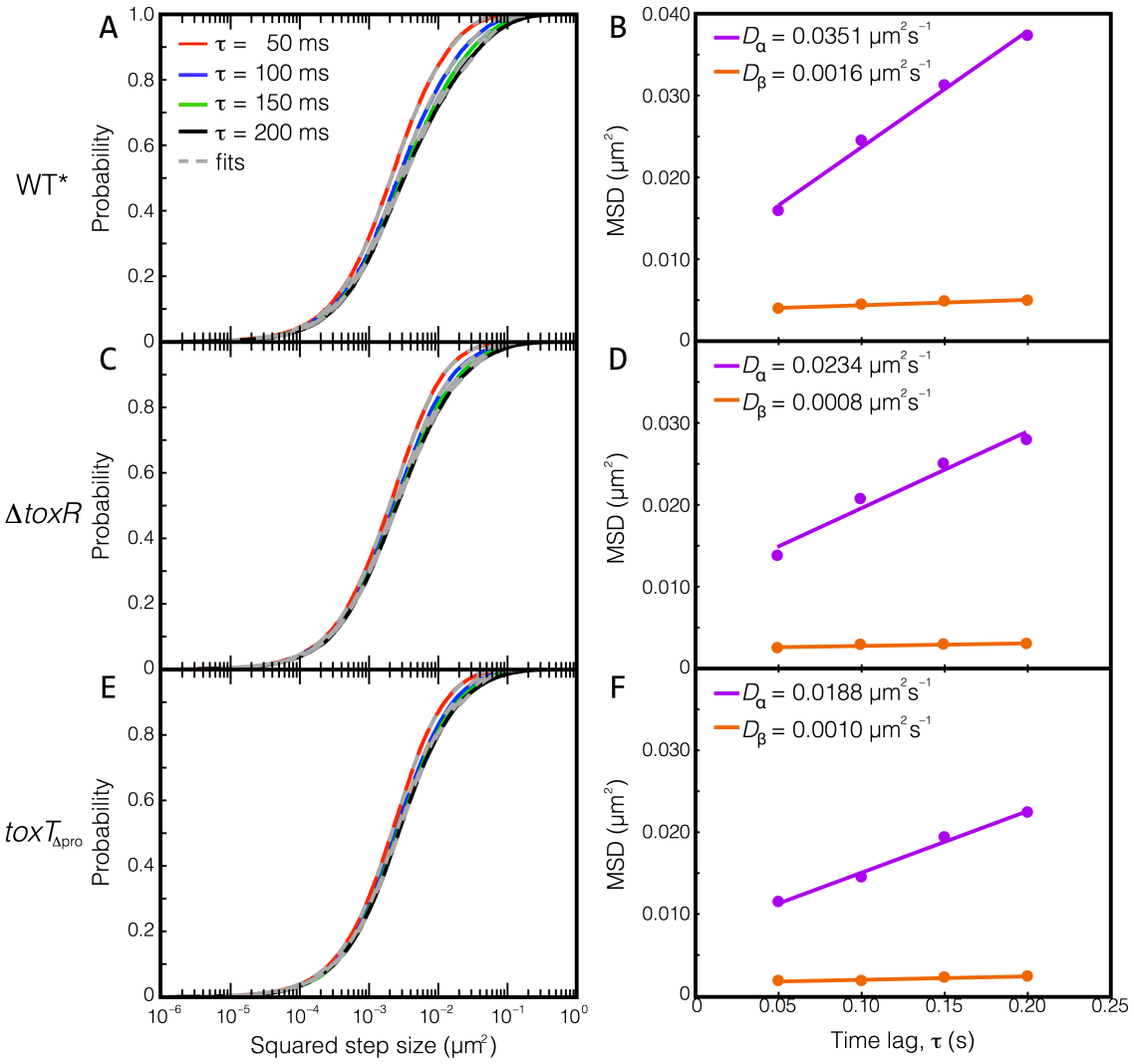

Figure 3.4: Cumulative probability distributions (CPDs) and population mean squared displacements (MSDs) for the three mutant strains

A, C, and E: Cumulative distributions of squared step size for each of the first four time lags (50, 100, 150 and 200 ms in red, blue, green, black, respectively) and best fit three-term CPD models (Eq. 3.3; 2 mobile populations and 1 immobile population; dashed gray lines).

B, D, and F: MSD vs. time lag, τ, for the fast and slow populations of each of the three strains examined. The slope of each curve is proportional to the fast diffusion coefficient, D_α, or the slow diffusion coefficient, D_β, respectively.

Table 3.4: Number of cells and trajectories studied
Steps taken in 50 ms (i.e. between two consecutive imaging frames) were used for the histograms in Fig. 3.1B, and the CPDs in Fig. 3.4. The standard error of the mean for all three strains was <0.005 nm.

Strain	Cells	Trajectories	50-ms steps	Mean step size (nm)
WT*	74	1083	21057	59.3
$\Delta toxR$	119	1144	22358	55.4
$toxT_{\Delta\text{pro}}$	54	2609	45225	56.8

Fig. 3.1B shows the normalized histogram of all steps (frame-to-frame displacements) taken in each of the three cell strains. At our 20 frames per second imaging rate, each step size indicates the displacement per 50 ms. In all three mutants, Fig. 3.1B shows a broad distribution of step sizes, consistent with the existence of several different types of motion, as described below. However, there are notable differences among the three cell types. In particular, TcpP takes more large steps (>100 nm) when both of its interaction partners are present (WT*) and smaller steps in their absence ($\Delta toxR$, $toxT_{\Delta\text{pro}}$; Fig. 3.1C). This is also reflected in the mean step size for each strain: the mean step size in WT* (59.3 nm) is greater than the mean step size in the knockout strains $\Delta toxR$ (55.4 nm) and $toxT_{\Delta\text{pro}}$ (56.8 nm). The numbers of cells, trajectories, and steps used in these histograms are given in Table 3.4. The standard error of the mean for all three strains was <0.005 nm.

In order to make comparisons among strains, we examined the CPDs for TcpP displacements in $\Delta toxR$ (Fig. 3.4C and D) and $toxT_{\Delta\text{pro}}$ (Fig. 3.4E and F). As was found for WT*, the knockout strains were well described by a three-term model (Eq. 3.3). The diffusion coefficients for the mobile populations differed among the strains, as did the amount each population contributed to the model (Fig. 3.5A; Table 3.2). Importantly, the diffusion coefficients for both the fast and slow populations are larger in WT* than in the knockout strains. Additionally, we compared the fraction of mobile molecules in the fast population to the fraction in the slow population (Fig. 3.5B) and found a greater contribution from the fast population in WT* than in either of the knockouts (fast fraction = 34.3%, 24.7%, and 31.3% for WT*, $\Delta toxR$, and $toxT_{\Delta\text{pro}}$, respectively).

3.3.5 An immobile TcpP population exists in all three mutant strains

In all three cell types examined, the best fit CPD (Eq. 3.3) included a significant immobile population, (Table 3.2; Fig. 3.5C). The immobile fraction is smaller, but not fully absent, when the *toxT* promoter is deleted (14.2% in $toxT_{\Delta\text{pro}}$ vs. 22.2% in WT*), indicating that TcpP binding to $toxT_{\Delta\text{pro}}$ accounts for some, but not all, of the observed

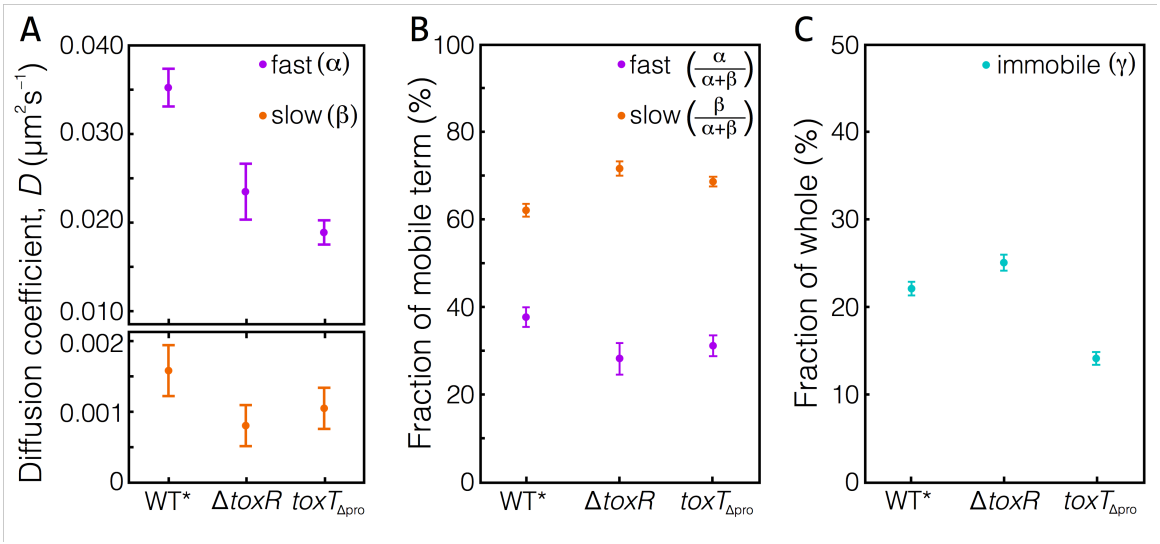

Figure 3.5: Diffusion coefficients and fractional contributions from CPD analysis
A: Diffusion coefficients for the fast and slow populations in all three strains, calculated from the MSD curves in Fig. 3.4. The bottom scale has been amplified to highlight differences among the values of the slow diffusion coefficient, D_β.
B: Contributions of the fast and slow populations to the mobile fraction for each strain.
C: Contribution of the immobile population to the total population for each strain examined.
All error bars ±1 standard deviation.

immobile population. We attribute some of this immobile population to imaging artifacts arising from capturing inherently three-dimensional motion (along the curved *V. cholerae* membrane) as a two-dimensional image. Movements perpendicular to the length of the cell—particularly those at the edges of the projection—appear shorter than movements in the longitudinal direction (Fig. 2.15A, green and pink curves, respectively). Monte Carlo simulations of molecules exhibiting Brownian diffusion on the surface of a cylinder indicate that curvature artifacts account for an immobile population of 2–6% of the model. However, after accounting for this artifact, a significant immobile TcpP-PAmCherry population remains, even in the absence of $toxT_{\Delta\text{pro}}$, implying that specific DNA-binding is insufficient to explain this immobile population.

3.4 Discussion

3.4.1 ToxR and *toxT*$_{\text{pro}}$ have similar impacts on TcpP diffusion

Considering that a TcpP–ToxR–$toxT_{\text{pro}}$ complex would be much larger than TcpP alone, one might suppose that TcpP would move faster on its own than when interacting with both ToxR and the *toxT* promoter. Instead, both the qualitative results from the step size histograms (Fig. 3.1B and C) and the quantitative results from CPD analysis (Fig. 3.5; Table 3.2) yield a surprising outcome: TcpP moves faster when it has the opportunity

to interact with both ToxR and the *toxT* promoter than when either partner is missing, and, even more notably, deletions of either ToxR or $toxT_{\mathrm{pro}}$ have similar impacts on TcpP diffusion. Additionally, when both ToxR and the *toxT* promoter are present, a larger fraction of mobile TcpP molecules are in the fast population.

Though a living cell is a complex system, the mutants examined in this chapter enable us to take a simplified view of the environment surrounding TcpP in *V. cholerae*. Because our 20 frames per second imaging conditions are selected to detect only the slowest TcpP molecules ($D < 4 \times 10^{-2}\,\mu\mathrm{m}^2/\mathrm{s}$; Kim et al., 2006), we attribute all of our trajectories to observations of the bitopic inner membrane protein TcpP that is interacting with the chromosome. As illustrated in Fig. 3.6A–D, respectively, TcpP may hop, slide and scan along DNA (Givaty and Levy, 2009); it may encounter nucleoid associated proteins (NAPs; Nye et al., 2000); it may bind nonspecifically to some DNA region; or it may bind specifically to the *toxT* promoter (Goss et al., 2010). Furthermore, TcpP might perform any of these motions while interacting with ToxR (Crawford et al., 2003; Fig. 3.6E–H).

Without $toxT_{\mathrm{pro}}$, TcpP should spend more time sliding and hopping along DNA (green box in Fig. 3.6) because there is no specific binding target to locate. As expected, we find a smaller fraction of immobile TcpP molecules in the promoter knockout strain ($toxT_{\Delta\mathrm{pro}}$, $\gamma = 14.2\%$) than in the wild type-like strain (WT*, $\gamma = 22.2\%$). Furthermore, comparing the contributions of the fast and slow populations (α and β) to the total mobile fraction ($\alpha + \beta$) in Fig. 3.5B, we see that the knockout strains ($\Delta toxR$ and $toxT_{\Delta\mathrm{pro}}$) are more similar to each other than to the wild type-like strain (WT*), despite the fact that there is only a single copy of the *toxT* promoter, while there are multiple copies of ToxR.

Overall, the large impact of ToxR on the TcpP diffusion coefficients implies that each TcpP molecule spends a significant amount of time interacting with a ToxR molecule. Previous work has shown that ToxR enhances the ability of TcpP to activate transcription of *toxT*: without ToxR, fewer modes of TcpP motion are possible (Fig. 3.6, red box), activation of *toxT* is greatly reduced, and only overexpression of TcpP can compensate (Higgins and DiRita, 1994; Häse and Mekalanos, 1998; Murley et al., 1999; Krukonis et al., 2000; Matson et al., 2007). It is likely then, that ToxR removes obstacles to TcpP diffusion toward its $toxT_{\mathrm{pro}}$ target.

In addition, if the TcpP–ToxR interaction were independent of the *toxT* promoter, we would expect deleting $toxT_{\mathrm{pro}}$ to have no effect on the dynamics or distribution of mobile TcpP molecules in comparing WT* and $toxT_{\Delta\mathrm{pro}}$. On the contrary, we observe that deleting $toxT_{\mathrm{pro}}$ has a similar effect on TcpP diffusion as deleting ToxR, indicating that binding to the promoter enhances the interaction between ToxR and TcpP.

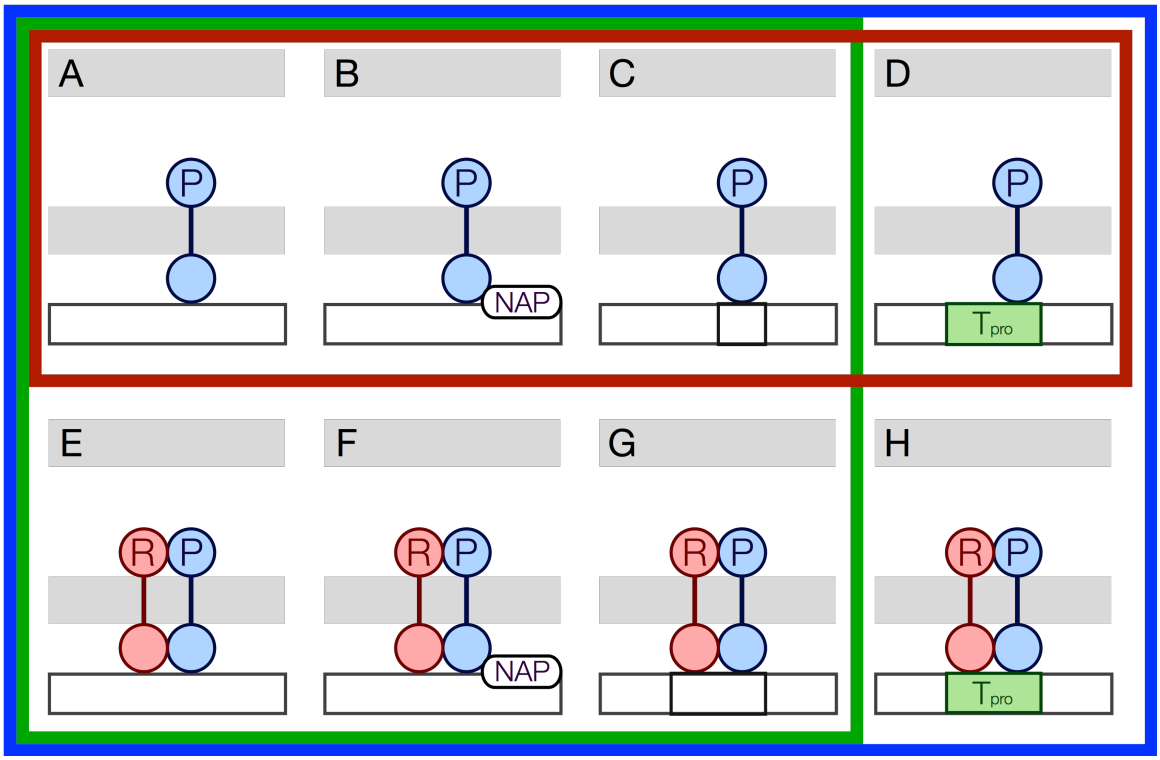

Figure 3.6: Expected modes of TcpP motion
A: TcpP (P) sliding along DNA (fast motion).
B: TcpP blocked by a nucleoid associated protein (NAP)
C: TcpP binding nonspecifically to DNA
D: TcpP bound specifically to the *toxT* promoter (T_{pro}; slow motion).
E–H: Modes A–D in the presence of ToxR.
Two additional modes of motion (free TcpP and TcpP interacting with ToxR without binding DNA) are too fast to be captured by our imaging setup (20 frames per sec). Of the modes shown, only interactions A–D can occur in the Δ*toxR* mutant (red box), and only modes A–C, and E–G are possible in the $toxT_{\Delta pro}$ strain (green box). All eight modes can occur in the WT* strain (blue box).

3.4.2 TcpP–ToxR–$toxT_{pro}$ interactions involve ToxR removing obstacles to TcpP diffusion

The mechanism by which TcpP regulates ToxT expression must be consistent with our findings that ToxR increases TcpP mobility, and that this increased diffusion speed relies also on the *toxT* promoter. The increase in diffusion speed in the presence of additional interaction partners suggests the involvement of nucleoid-associated proteins (NAPs), which bind large segments of bacterial DNA, compacting the chromosome and preventing excessive transcription (Dame et al., 2011; Wang et al., 2011). In *V. cholerae*, H-NS is a small, abundant NAP that binds the *toxT* promoter, as well as the *ctx* and *tcpA* promoters (Nye et al., 2000) and represses transcription at these sites. ToxT counteracts H-NS repression of *ctx* and *tcpA* (Nye et al., 2000; Yu and DiRita, 2002; Nye and Taylor, 2003), likely by displacing H-NS from the DNA (Stonehouse et al., 2011), and ToxR may do the same for *toxT* (Morgan et al., 2011). Indeed, in the absence of H-NS, neither TcpP nor ToxR are required for ToxT expression (Nye et al., 2000). If this is the case, ToxR may increase TcpP movement along DNA by removing the obstructing proteins; without ToxR, TcpP diffusion would be hindered.

The results of our CPD analysis are consistent with a mechanism (Fig. 3.7) in which ToxR molecules remove occluding NAPs such as H-NS from the chromosome, then locate and bind to the *toxT* promoter, where they recruit TcpP. Together, the ToxR–TcpP complex activates *toxT* transcription. Afterward, the proteins may continue along the DNA together, with ToxR continuing to clear the way of NAPs. Furthermore, our observed trends suggest that, when ToxR is not present the progress of TcpP is reduced, perhaps by the presence of H-NS, which makes it more difficult for transcription of *toxT* to occur. Overexpression of TcpP might restore *toxT* activation by increasing the chances of any particular TcpP molecule to find the promoter (Higgins and DiRita, 1994; Häse and Mekalanos, 1998; Murley et al., 1999; Krukonis et al., 2000; Matson et al., 2007). The absence of $toxT_{\mathrm{pro}}$ has a similar impact on TcpP diffusion as the absence of ToxR: ToxR is not able to bind, and thus cannot recruit TcpP. TcpP then slowly moves along the DNA, almost as if ToxR were not present, faced with myriad DNA-binding obstacles.

3.5 Conclusions

The unusual membrane-bound transcription mechanism of the ToxR regulon remains key to understanding the virulence pathway in *V. cholerae*. Using super-resolution microscopy and single-molecule tracking with better than 40 nm resolution, we have examined the motions of TcpP, an essential protein in the ToxR regulon. We find that TcpP labeled with PAmCherry retains its ability to function and localizes correctly to

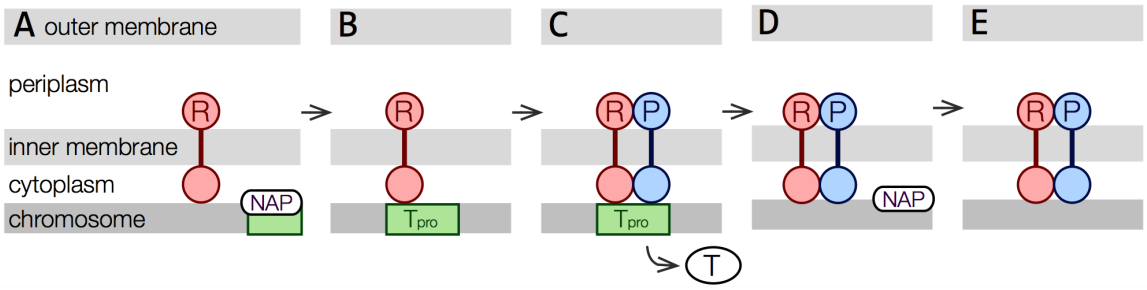

Figure 3.7: Mechanism of TcpP–ToxR–toxTpro interaction
ToxR binds DNA and scans along it, removing a nucleoid associated protein (NAP) such as H-NS as it travels (A). When it finds the *toxT* promoter (B), ToxR recruits TcpP, which activates ToxT transcription (C). After transcription, the proteins may continue together (D), with ToxR continuing to clear the way (E).

the inner membrane. Comparisons of TcpP-PAmCherry step sizes and of the cumulative probability distributions of these steps taken in WT* cells and cells without either the TcpP binding partner ToxR or the $toxT_{\mathrm{pro}}$ DNA binding region reveal that TcpP moves faster when both ToxR and $toxT_{\mathrm{pro}}$ are present than when either is removed. Furthermore, a significant population of immobile TcpP molecules exists in all three strains, even when the *toxT* promoter has been removed and specific DNA-binding cannot occur. The evidence supports a mechanism in which ToxR recruits TcpP to $toxT_{\mathrm{pro}}$; when ToxR is removed, TcpP is less likely to find the promoter, and when $toxT_{\mathrm{pro}}$ is removed, TcpP is less likely to encounter ToxR.

Though rare, membrane-bound transcription activation is not restricted to *V. cholerae*; indeed similar mechanisms have been observed in several other organisms (Table 1.1; Kolibachuk and Greenberg, 1993; Dell et al., 1994; D'Elia and Salyers, 1996; Lassak et al., 2013). Investigating the specific role of TcpP in the *V. cholerae* virulence pathway will then also shed light on the mechanisms and dynamics of membrane-bound transcription regulation more generally. Future experiments labeling ToxR or the DNA near the *toxT* promoter will enable dual-color imaging of this transcription activation mechanism, lead to greater understanding of the role of NAPs in the regulation of transcription, and offer additional insight into interactions on the molecular scale. The mechanism by which the membrane localized complex recruits RNA polymerase remains a significant gap in our understanding of how such activators stimulate transcription, and our studies are aimed at uncovering this process.

Chapter 4

Biocompatible plasmonic substrates for enhanced single-molecule fluorescence in live bacteria[1]

4.1 Introduction

Single-molecule super-resolution fluorescence microscopy can achieve resolutions well below the ∼300-nm diffraction limit of visible light. The brighter the fluorophore used, the better the precision with which it can be localized (Thompson et al., 2002), and the better the photostability of the fluorophore, the longer the probe can be imaged and tracked. Using organic dyes *in vitro*, single-molecule microscopy can reach 1.5-nm localization precision (Yildiz et al., 2003), but in live bacteria, where dimmer, less photostable fluorescent proteins are the fluorophores of choice and cellular autofluorescence produces a high background level of fluorescence, the typical localization precisions achieved are an order of magnitude larger (10–40 nm; Bates et al., 2007; Biteen et al., 2008). Enhancement from localized surface plasmon resonance offers a way to improve both the brightness and the stability of fluorescent proteins in live cells.

A localized surface plasmon is the collective oscillation of electrons in the conduction band of a metal nanoparticle (Xia and Halas, 2005; Willets and Van Duyne, 2007). This plasmon mode has a particular resonance frequency, which depends on the size, shape and material of the nanoparticle. For gold and silver nanoparticles, this resonant wavelength can be tuned across the visible region of the spectrum (390–700 nm; Xia and Halas, 2005; Zhang et al., 2005). When the nanoparticle interacts with light at the resonant wavelength, the plasmon mode is excited, which creates a concentrated electric field confined to a small volume near the metal surface; this can act as an antenna for a fluorophore of a matching excitation or emission wavelength that is placed in this field (Willets and Van Duyne, 2007; Taminiau et al., 2008). The excitation cross-section of

[1] In collaboration with Jessica Donehue, who prepared and characterized the nanostructured gold substrates.

the fluorophore in this local field is enhanced, leading to increased brightness, and the radiative decay rate is increased, reducing the likelihood of photobleaching and enhancing the stability of the fluorophore (Anger et al., 2006; Fu et al., 2008). In addition to enabling greater localization precision, a brighter fluorophore allows a sample to be imaged at faster frame rates without losing signals in the background noise. More stable, longer-lived fluorophores also make longer trajectories possible. In live cells, these advantages would lead to substantial improvement in the spatial and temporal resolution of molecular-scale dynamics.

Because the strong electric field decays rapidly as a function of distance from the nanoparticle surface, plasmon enhancement is highly distance-dependent (Fig. 1.6; Anger et al., 2006). Significant fluorescence enhancement occurs when fluorophores are located within 50 nm of the plasmonic nanoparticle. At very small distances (<3 nm), quenching of fluorescence occurs due to an increased non-radiative decay rate and electron transfer. We propose a sample geometry in which bacterial cells are mounted on top of a microscope coverslip that has been patterned with nanostructured gold (Fig. 4.1). *In vitro* experiments generally rely on an insulating spacer layer to prevent quenching (Jose et al., 2013), but this 3–50 nm separation distance requirement is well suited to imaging fluorescent proteins in the membrane or periplasm of live bacteria: fluorophores in and near the cell membranes are positioned well within the enhanced field when the cells are placed directly on a coverslip coated with metal nanoparticles. For example, the entire cellular envelope of *Vibrio cholerae* bacteria is less than 50 nm thick (Graham et al., 1991), so fluorescent proteins located in the periplasm are near enough to be enhanced by a plasmonic substrate without being close enough for quenching to occur (Fig. 4.1).

Fluorescent proteins are highly specific labels because they are genetically encoded as fusions to proteins of interest. They can thus be produced directly by the cell, and, unlike small molecule dyes, they do not require additional steps to be incorporated into or washed out of the cell (Chen and Ting, 2005; Giepmans et al., 2006; Fernández-Suárez and Ting, 2008). Since the determination of the green fluorescent protein (GFP) crystal structure (Ormö et al., 1996), a wide variety of fluorescent proteins have been developed (Chudakov et al., 2010). This diversity of fluorescent probes includes photoactivatable proteins that undergo conformational changes to switch from dark, non-fluorescent states to fluorescent states upon exposure to violet light (Subach et al., 2009a,b), which are suitable for Photoactivated Localization Microscopy (PALM) and related techniques such as Fluorescence Photoactivation Localization Microscopy (FPALM) and Stochastic Optical Reconstruction Microscopy (STORM; Betzig et al., 2006; Rust et al., 2006; Hess et al., 2006). These probes are also widely used for single-molecule tracking in live cells. Multiple tracking experiments can be carried out in a single cell by activating a small number of fluorophores at a time.

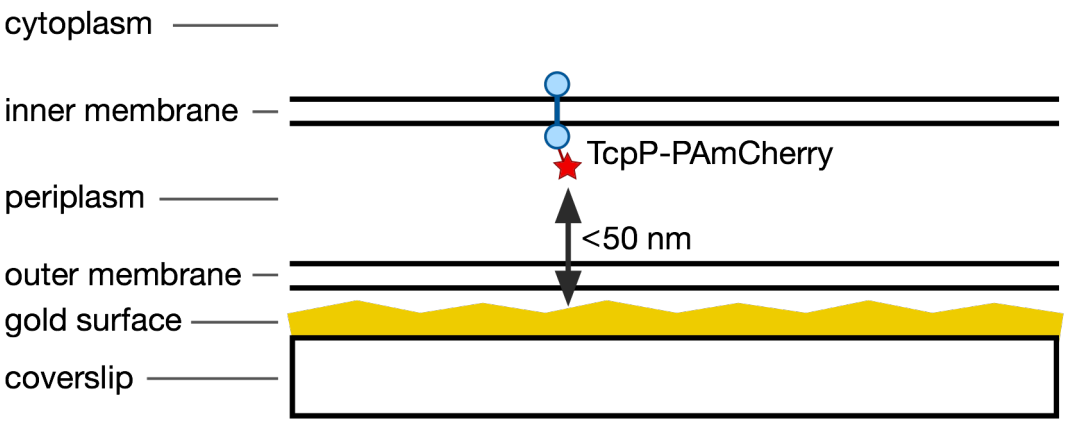

Figure 4.1: Cutaway view of a *V. cholerae* cell on a patterned gold substrate
The cellular envelope is less than 50 nm thick. TcpP labeled with PAmCherry resides on the inner membrane, near enough for enhancement from the gold plasmon mode.

Autofluorescence can overwhelm the signal from a fluorescent protein in cell, particularly in the green region of the spectrum (∼550 nm; Benson et al., 1979). Red fluorophores can be detected better over this background, but the quantum yield (QY) for red fluorescent proteins remains quite poor (e.g. mCherry has a QY of 0.22; Shaner et al., 2004). Previous *in vitro* work in our lab has shown that the brightness and photostability of the red fluorescent proteins mCherry and PAmCherry can be enhanced by gold nanorods (Donehue et al., 2013, 2014).

Although both silver and gold are commonly used for plasmon enhancement experiments in the visible spectrum (Xia and Halas, 2005; Zhang et al., 2005), we chose to use gold for its resistance to oxidation and to ensure biocompatibility. We used two types of patterned gold substrates: an annealed nanoisland film, and a periodic array of nanotriangles created using nanosphere lithography.

In this chapter, we develop protocols for enhancing mCherry and PAmCherry fluorescence in live *V. cholerae* cells based on coupling to patterned gold substrates. We investigate *V. cholerae* bacteria expressing fusions of the membrane-bound transcription activator protein TcpP to either mCherry or PAmCherry. TcpP is a key protein in the *V. cholerae* pathogenic pathway, activating transcription of ToxT, which in turn actives transcription of the toxin responsible for the major symptoms of the human disease cholera (See Chapter 3). TcpP localizes to the inner membrane, with a cytoplasmic DNA-binding region at its N-terminus, and a periplasmic C-terminus (Carroll et al., 1997; Häse and Mekalanos, 1998). The fluorescent protein labels are fused to this C-terminal end to avoid disrupting DNA binding (Fig. 3.1A). By improving the observed photophysical

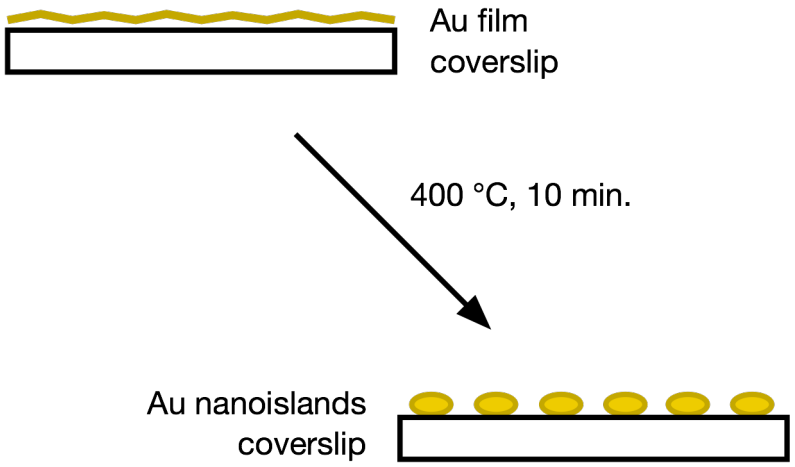

Figure 4.2: Procedure for synthesizing gold nanoisland substrates
The initial gold film is deposited onto the coverslip without a wetting layer. Annealing the film causes the gold to cluster into nanoislands (Warmack and Humphrey, 1986).

properties of fluorescent proteins in live bacteria using plasmonic enhancement, we have therefore developed the tools to answer biological questions with high precision.

4.2 Methods

4.2.1 Plasmonic substrates

4.2.1.1 Gold nanoisland films

Gold nanoisland films were prepared by evaporating 6 nm of gold onto glass coverslips (24×40 mm) under vacuum (5×10^{-6} Torr; Denton Vacuum SJ-20 evaporator). The coverslips were annealed at 400°C for 10 min. Annealed films exhibited a color change from blue to pink. Since the gold was deposited directly onto the coverslip without a wetting layer, this thin layer of gold on glass clusters together to form nanoislands when annealed (Fig. 4.2; Warmack and Humphrey, 1986).

4.2.1.2 Gold nanotriangle arrays

Glass coverslips were coated with 100 nm ITO by sputter deposition (Kurt J Lesker Lab 18-1), then oxygen plasma-etched for 15 minutes at 200 mTorr (PlasmaEtch PE-50). 750-nm polystyrene beads (Polysciences Polybead Microspheres) were centrifuged (10 min, 25 °C, $12,000 \times g$) and washed three times: twice with 1 mL 95% ethanol, and once with a solution of 0.5 mL 95% ethanol and 0.5 mL 0.2 mM sodium dodecyl sulfate (SDS). Plasma-etched coverslips were placed in a polystyrene Petri dish and covered with 0.2 mM

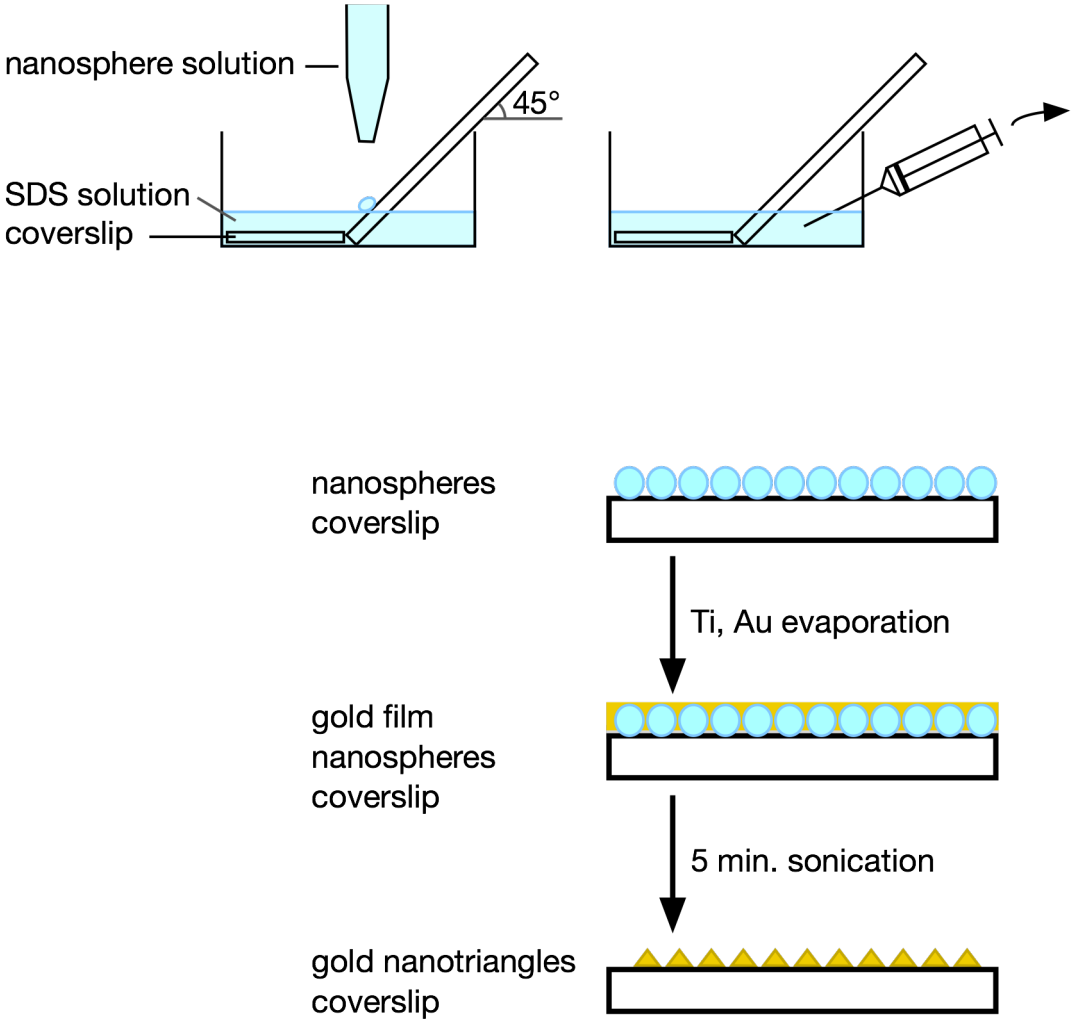

Figure 4.3: Nanosphere lithography procedure
After the SDS solution is removed, the sample is allowed to air-dry before metal evaporation.

SDS. A microscope slide was placed in the Petri dish at a 45° angle such that one end rested at the edge of the coverslip (Fig. 4.3). The beads were pipetted onto the solution surface 3 µL at a time until the surface appeared opalescent, indicating a monolayer of beads were present (30–40 µL total). The SDS solution was removed via a syringe inserted into the side of the Petri dish, and the polystyrene beads self-assembled into a close-packed monolayer (Tan et al., 2005). Coated coverslips were allowed to dry at room temperature for several hours.

5 nm titanium was evaporated onto the nanosphere-coated coverslip as a wetting layer, and 50 nm of gold was evaporated on top of the titanium at a deposition rate of 1 nm/s under vacuum (1×10^{-6} Torr; Denton Vacuum SJ-20 evaporator). Coverslips were soni-

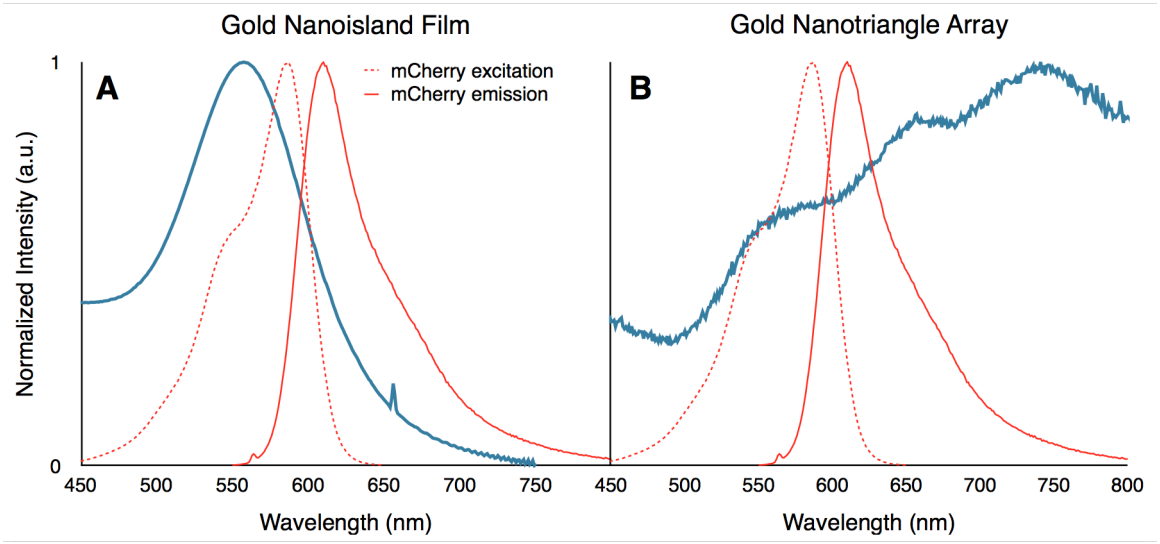

Figure 4.4: Spectra of gold substrates
A: UV-vis spectrum of a gold nanoisland film (blue).
B: Dark-field scattering spectrum of a gold nanotriangle array (blue).
Exciation and emission spectra for mCherry are overlaid on the spectra for each substrate.

cated in a toluene bath for 5 minutes to remove the nanosphere mask, leaving behind a periodic array of nanoscale gold triangles.

4.2.1.3 Plasmon resonance characterization

UV-visible absorption spectroscopy was used to determine the resonance wavelengths of the gold nanoisland films (Hewlett Packard 8453; Fig. 4.4A). The resonance wavelength of each gold nanotriangle array was determined using dark-field scattering spectroscopy (Fig. 4.4B). A broadband halogen light source (400–1000 nm) was used to excite the nanotriangle samples through a dark-field oil-immersion objective with an NA of 0.6. The samples were immersed in water and a diffraction-limited spot from a nanotriangle was aligned to the entrance slit of a Princeton Instruments Acton2300 imaging spectrograph. Data were collected on an Andor iXon EMCCD detector. Background spectra taken from an area of the sample without the gold surface were subtracted from the measured gold scattering spectrum. To correct for the spectral efficiency of the system, the spectra were divided by the detected spectrum of the broadband light source.

4.2.2 Bacterial samples

4.2.2.1 Bacterial cultures

V. cholerae O395 Δ*tcpP* TcpP-PAmCherry O395 Δ*tcpP* TcpP-mCherry bacterial cells were grown in LB media at 37 °C, then diluted into M9 minimal medium and grown to

log phase (OD ∼0.3) at 30 °C as described in Chapter 3. To induce expression of the TcpP-(PA)mCherry fusion proteins, the cultures were incubated in arabinose (0.10% final concentration) for 3 hours at 30 °C before fixation or live sample preparation.

4.2.2.2 Bacterial fixation

5-mL aliquots of *V. cholerae* culture were centrifuged 5 minutes at 4 °C and 4000 rpm to pellet. The supernatant was discarded, and the cell pellet was resuspended in 5 mL of a refrigerated solution of 0.67 *M* methanol in M9. The cells were incubated at room temperature (∼25 °C) for 10 minutes, then on ice for 30 minutes. The samples were then centrifuged for 15 min at 4 °C and 7830 rpm. The supernatant was discarded and the pellet was resuspended in 5 mL cold M9. Centrifugation and resuspension were repeated twice more for a total of three washes. Fixed cells were kept chilled (4 °C) until use (at most two days), at which time a 1-mL aliquot of fixed cells was centrifuged for 5 minutes at 7830 rpm to pellet, then resuspended in ∼40 µL residual supernatant to concentrate.

4.2.2.3 Live cell preparation

A 1-mL aliquot of live cells was centrifuged for 30 s and $30,000 \times g$ at room temperature to pellet. The supernatant was discarded and the pellet was resuspended in 1 mL warm M9 (30 °C). Centrifugation and resuspension were repeated for a total of two washes. After the final centrifugation, the pellet was resuspended in ∼40 µL of residual supernatant to concentrate the cells.

4.2.3 Microscope sample preparation

2.0-µL droplets of live or fixed cells expressing TcpP-(PA)mCherry were pipetted onto an agarose pad (2% agarose in M9 minimal medium) prepared on a microscope slide. The cell droplets were then covered with a coverslip that was patterned with a gold nanoisland film or gold nanotriangle array, with the patterned side of the coverslip facing the cells. Each patterned coverslip was prepared such that one reference region was masked from gold evaporation, and care was taken to place at least one cell droplet in a region that would be covered by a portion of the coverslip that lacked the gold pattern (Fig. 4.5). This served as a built-in control: cells with and without the gold substrate experienced otherwise identical imaging conditions. Excess agarose was trimmed away before samples were imaged. Only cells expressing TcpP-PAmCherry were imaged on the nanotriangle arrays.

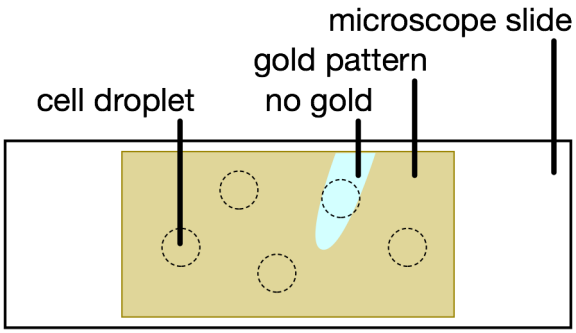

Figure 4.5: Sample geometry for cells on gold substrates
Several droplets of cells were used to enable multiple experiments in a single sample.

4.2.4 Super-resolution microscopy

Samples were imaged for 7–10 minutes at room temperature using wide-field epifluorescence microscopy through a 100× oil immersion phase contrast objective (NA 1.40; Zeiss Immersol 518F immersion oil) on an Olympus IX71 inverted microscope and a Semprex micrometer stage. A PIFOC piezo element was used for precision focusing. A Photometrics Evolve EMCCD camera with >90% quantum efficiency was used to detect emission from the sample at 25 frames per second (40 ms integration time), with each pixel corresponding to a 49×49 nm area on the sample. Fluorescence of mCherry and PAmCherry was excited using 561-nm laser light (0.2–0.3 µW/µm^2; Coherent Sapphire 560-50). Subsets of PAmCherry molecules were activated into a fluorescent state using coaligned 405-nm laser light (0.006–0.2 µW/µm^2; Coherent Cube 405-100). The excitation and activation lasers were coupled using a dichroic mirror (Semrock Di01-R405) and filtered with appropriate excitation, emission and dichroic mirrors (Semrock LL01-561, Semrock BLP01-561, Semrock Di01-R561). Both laser beams were circularly polarized (Tower Optical AO15Z 1/4 556, Tower Optical AO15Z 1/4 408). A pair of shutters (Uniblitz) were use to control the laser beams such that a sample was exposed to only one laser beam at a time (Fig. 2.7). Cells were exposed to 405-nm light for 50–70 ms every 70-90 s for 7 minutes of data collection per movie.

4.2.5 Background fluorescence subtraction

Background fluorescence due to gold scatter was measured by averaging the last 50 frames of a movie, after all fluorescent proteins had bleached and only fluorescence from the gold substrate remained. This background intensity was subtracted from each imaging frame of the movie. Background-subtracted frames were used for further analysis.

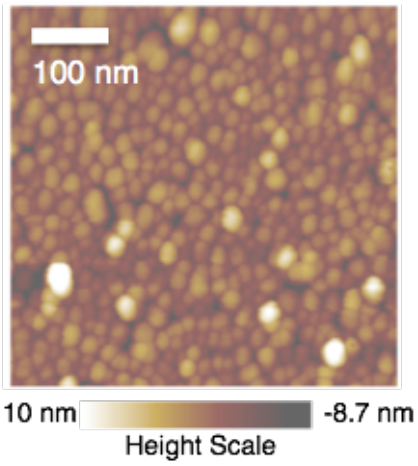

Figure 4.6: Tapping mode AFM scan of gold nanoislands.
Nanoislands are 10–12 nm tall and 30–40 nm in diameter on average.
Figure from Donehue et al., 2013.

4.3 Results

4.3.1 Gold nanoisland films

Nanoisland films were characterized using atomic force microscopy (AFM; Agilent PicoPlus 5500 Atomic Force Microscope; Fig. 4.6).

The gold nanoisland substrates have good spectral overlap with mCherry and PAmCherry excitation (Fig. 4.4A), however, the films had very high background scattering intensity that also fluctuated significantly during imaging, leading to imperfect background subtraction.

Furthermore, cells were difficult to locate in bright field on some films due to the density of nanoislands. Because mCherry does not require activation to be fluorescent, it was much more straightforward to find cells expressing TcpP-mCherry than those expressing TcpP-PAmCherry on nanoisland films. During the first few seconds of imaging a region of the sample, the cell outline was visible above the background scatter from the gold. The initial fluorescence emission collected came from a large number of molecules clustered too densely for super-resolution localization, but the density decayed to single-molecule levels as mCherry molecules bleached.

Fluorescence from cells expressing TcpP-PAmCherry was much more difficult to detect on the dense gold nanoisland films. PAmCherry was initially in a non-fluorescent state, and the partial bleaching strategy used for TcpP-mCherry imaging could not be used for the photoactivatable protein. Without clear cell locations in bright field, and with very low levels of fluorescence under laser illumination, the TcpP-PAmCherry *V. cholerae* cells were essentially invisible.

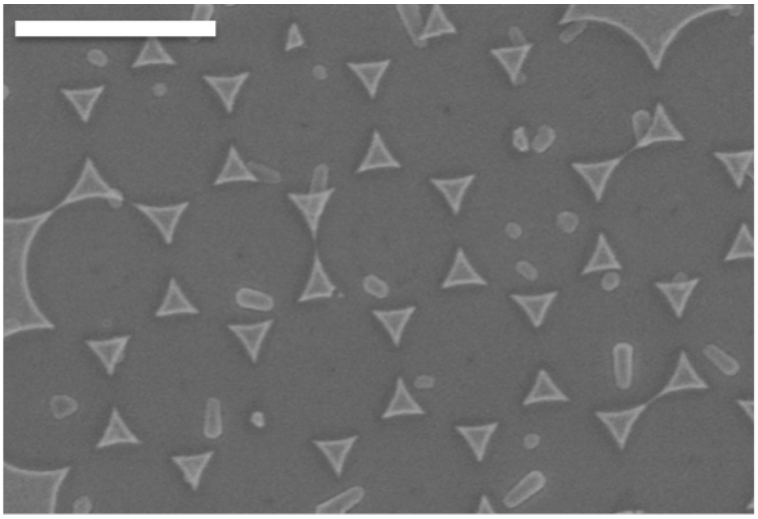

Figure 4.7: Scanning electron micrograph of a nanotriangle array
The tip-to-tip distance for the nanotriangles is approximately 150 nm. Nanotriangles are roughly 250 nm on a side. Scale bar is 1 µm.

In addition to the imaging challenges, the nanoisland films presented problems with sample durability. Gold does not attach to glass well without a wetting layer, which allows the thin films to form nanoislands (Warmack and Humphrey, 1986), but this property of the material also makes the samples fragile. Nanoislands flaked off the glass coverslip easily. Mere contact with the agarose gel used to keep the cells immobilized and hydrated was often sufficient to transfer nanoislands from the coverslip to the agarose surface.

4.3.2 Gold nanotriangle arrays

Scanning electron microscopy (Hitachi SU8000) was used to verify the formation of gold triangles on the coverslip surface in an ordered array approximately 250 nm on a side, with a tip-to-tip distance of approximately 150 nm (Fig. 4.7). The periodic gold pattern was also visible using diffraction-limited phase-contrast microscopy (Fig. 4.8).

Cells were easily distinguishable on the gold pattern, particularly using phase contrast (Fig. 4.8B). Background scatter from the nanotriangles was low intensity and stable during imaging. Fluorescence from TcpP-PAmCherry was detectable above the background. After background subtraction, single-molecule fluorescence was clearly distinguishable (Fig. 4.8C).

Nanotriangle substrates were relatively durable. We were able to reuse substrates by removing the agarose pad, cleaning the gold-coated coverslip with Amphyl disinfectant and rinsing in water and acetone to remove the cells and microscope immersion oil. A second oxygen plasma etching was also used to clean the substrate, though this weakened the coverslip. This durability makes the nanotriangle substrates much better than the

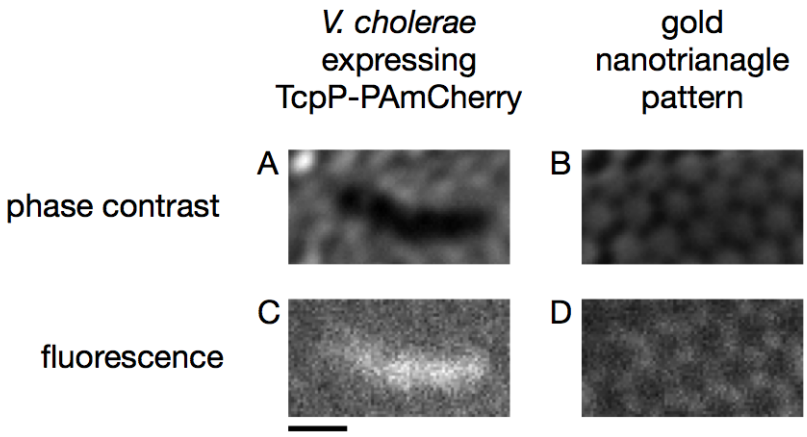

Figure 4.8: *V. cholerae* on gold nanotriangles
A: Phase contrast image of a single *V. cholerae* cell on a nanotriangle substrate.
B: Phase contrast image of a nanotriangle substrate.
C: Background-subtracted fluorescence from TcpP-PAmCherry in the cell shown in A (50-frame sum).
D: Background scatter from nanotriangle array (50-frame average).
Scale bar is 1 µm.

nanoisland films for looking at plasmon-enhanced fluorescence from labeled proteins in live bacteria cells.

4.3.3 Fixed-cell studies

We used fixed *V. cholerae* in order to count the photons emitted by immobile molecules in the cells on the gold nanoisland substrates and cells on glass as a measure of fluorescence enhancement. The inconsistent background scatter of the nanoislands could not be accounted for in our background subtraction procedure, making it difficult to distinguish fluorescence emission originating in the cells from fluctuations in substrate scatter. Fluorescent molecules also appeared to diffuse in some fixed cells. This may be attributed to the fixation protocol used. Tanaka et al. (2010) have reported that membrane-bound molecules in human T24 cells retain mobility in cells fixed using a cold methanol wash, despite the cross-linking that is intended to preserve the cells and immobilize the molecules. Fixed cells were also slightly more difficult to see in bright field than live cells. Because of these factors, we chose to image only live cells on the nanotriangle substrates.

4.4 Discussion

The major advantages of the nanoisland substrates are the density of potential hotspots for plasmon enhancement, and the simplicity of the substrate preparation. The high density of nanoislands is also a major disadvantage, however, since it was more difficult

to see bacteria through the denser films. Coverslips coated with gold nanotriangles were more sparse than the nanoislands, but cells could be located easily using a phase contrast objective. Nanotriangle arrays are also relatively simple to create, they are more consistent from sample to sample, and they are durable enough to be used multiple times.

There are few examples of coupling fluorescence from biological samples to plasmonic materials in the literature to date: plasmonic enhancement has been observed in model membranes and eukaryotic cells, with enhancements of 4–10-fold (Levene et al., 2003; He et al., 2006; Le Moal et al., 2007; Lohmüller et al., 2011; Hao et al., 2012). To our knowledge, this chapter is the first protocol for plasmonic enhancement of fluorescence in live bacteria. The existing studies largely rely on silver substrates (He et al., 2006; Le Moal et al., 2007; Hao et al., 2012) for plasmonic enhancement, but silver has long been used for its antimicrobial properties (Arvizo et al., 2012), so it is not likely to be compatible with our bacterial samples.

In *in vitro* experiments from our lab, gold nanoisland films increased mCherry brightness by up to fourfold, and lifetime until photobleaching by up to sevenfold (Donehue et al., 2013). We anticipate similar enhancements for PAmCherry in the live *V. cholerae* cells.

4.5 Conclusions

In this chapter, we have presented a protocol for achieving plasmon-enhanced fluorescence in live *V. cholerae* bacteria using patterned gold substrates. We found that single-molecule fluorescence from TcpP-mCherry and TcpP-PAmCherry could be detected above the background scatter of the gold nanoisland and nanotriangle surfaces. Additionally, we found that gold nanotriangle arrays prepared using nanosphere lithography are better suited for imaging single fluorescent molecules in bacteria than the nanoisland films. Cells are more visible in bright field images, the background scatter is easily subtracted, and the substrates are more consistent in feature density than the annealed gold films. The nanotriangle size and spacing (and thus the plasmon resonance) can be tuned by using polystyrene beads of varying sizes. Further investigations are needed to verify cell viability, but exposure to the gold nanotriangles did not noticeably harm the bacteria: cells on the gold substrate did not appear significantly different from cells on glass. With a very small sample size, we cannot conclusively quantify the degree of enhancement at this time. Future studies will enable us to quantify enhancement of mCherry and PAmCherry expressed in live cells and ultimately achieve more precise measurements of protein localization and dynamics in bacteria.

Chapter 5

Conclusions and Future Directions

In this thesis, we developed the experimental capabilities to examine individual protein molecules at precisions better than the diffraction limit of light. In particular, we observed, mapped and tracked the protein TcpP–an important regulator of virulence in the cholera disease pathway–in live *Vibrio cholerae* bacteria cells to understand how TcpP searches for and binds to its DNA target, the *toxT* promoter ($toxT_{\mathrm{pro}}$), using single-molecule fluorescence localization microscopy. In addition, we have developed a protocol for coupling fluorescently tagged *V. cholerae* membrane proteins to nanostructured gold substrates for plasmon-enhanced fluorescence.

5.1 Summary

In Chapter 2, we described several important methods implemented and obstacles encountered in single-molecule tracking in live bacteria cells, with particular examples from our work designing the TcpP diffusion measurements detailed in Chapter 3. We found that the choice of labeling schemes is very important: for example, the TcpP-Dendra2 fusion protein is an unstable construct, while the TcpP-mCherry fusion is not photostable enough for single-molecule tracking experiments, though its blinking, or "kindling," behavior is an advantage for PALM super-resolution imaging. Live-cell super-resolution bacterial microscopy is a growing field that is reaching beyond model systems, e.g. *Escherichia coli* and *Caulobacter crescentus*, and we discussed extensions to medically relevant organisms like *V. cholerae*.

We applied the labeling, imaging, and analytical methods described in Chapter 2 to determine the dynamics of the membrane-bound transcription activator protein TcpP in live *V. cholerae* bacteria in Chapter 3. We found that TcpP exhibits three populations of molecules: a population of relatively fast diffusing molecules, one of slower diffusing molecules, and a third that appears immobile within our ∼40 nm localization precision. Based on knockout studies, we further determined that TcpP-PAmCherry moves faster in the presence of its binding partner ToxR and its DNA target $toxT_{\mathrm{pro}}$ than when either partner is missing. We found that ToxR and $toxT_{\mathrm{pro}}$ deletions also have similar impacts

on TcpP diffusion, despite the difference in copy number. Our evidence supports a TcpP–ToxR–$toxT_{\mathrm{pro}}$ interaction mechanism in which ToxR recruits TcpP to the promoter to activate transcription of *toxT*.

In Chapter 4, we sought to improve the brightness and photostability of fluorescently labeled TcpP in live *V. cholerae* for live-cell imaging. To achieve this, we developed a method to couple PAmCherry-labeled TcpP to gold surfaces for plasmon-enhanced fluorescence. We found that an array of gold nanotriangles created using nanosphere lithography was better suited to imaging bacterial samples than annealed gold nanoisland films due to its durability and sample-to-sample consistency. TcpP-mCherry and TcpP-PAmCherry foci were detectable above the background caused by gold scatter.

5.2 Outlook

In this thesis, we have examined a single component of the ToxR regulon, and though TcpP diffusion is important for understanding membrane-bound transcription activation of *toxT*, there remains much more to learn. Since we found TcpP motion to be strongly influenced by ToxR, single-molecule studies of ToxR would complement our TcpP-tracking results. Colocalization studies of TcpP with its binding partners would further elucidate aspects of the regulon mechanism. By labeling ToxR with a fluorophore of a second color, e.g. the yellow fluorescent protein mCitrine, we could observe TcpP and ToxR diffusion simultaneously and identify interactions directly. We would expect to see TcpP and ToxR colocalized both when mobile (e.g. scanning DNA) and immobile (e.g. activating *toxT* transcription). However, ToxR has other roles in the cell (Krukonis et al., 2000; Li et al., 2000; Morgan et al., 2011; Goss et al., 2013), which would complicate tracking of ToxR as part of the regulon. Similarly, labeling the DNA upstream of *toxT* using the *lac* repressor system would reveal binding and sliding interactions between TcpP and $toxT_{\Delta\mathrm{pro}}$. Colocalization of TcpP and *toxT* would enable us to distinguish specific and nonspecific binding events. Labeling the chromosome would be a challenge: the label must have a small number of copies very close to *toxT* to identify the position of the *toxT* promoter clearly, but it should not be so close as to interfere with transcription activation or binding by TcpP and ToxR.

Beyond colocalization, there are other modifications we can make to understand the interplay among regulon components. Though TcpP and ToxR both bind to the *toxT* promoter region, they do not bind at precisely the same sites. Modifying the promoter to prohibit binding of one protein but not the other might disrupt the recruitment mechanism, which could be detected by changes in TcpP diffusion similar to the $\Delta toxR$ and $toxT_{\Delta\mathrm{pro}}$ strains used in Chapter 3. Alternatively, we could use TcpP or ToxR mutant proteins that are unable to bind DNA. The TcpP and ToxR binding sites could also be separated, preventing a potential TcpP–ToxR complex from binding to $toxT_{\mathrm{pro}}$ when as-

sembled. Conversely, a TcpP and ToxR mutant in which the two proteins are cross-linked could provide a diffusion coefficient for the TcpP–ToxR complex, and further clarify the populations present.

As our proposed mechanism relies on the presence of nucleoid-associated proteins like H-NS, the influence of such proteins on TcpP diffusion could be examined by controlling their expression. Knocking out H-NS entirely might prove problematic, since it regulates much more than *toxT* expression (Nye et al., 2000; Dame, 2005; Ghosh et al., 2006; Stonehouse et al., 2011), but we would expect that overexpression of H-NS would be similar to the removal of ToxR: TcpP diffusion would be hindered greatly, even in the presence of ToxR (as in WT*). Without ToxR ($\Delta toxR$), overexpressing nucleoid associated proteins would limit TcpP diffusion even further.

With continued development, we anticipate that plasmonic substrates can yield improvements in the amount of information available in live cell super-resolution microscopy experiments, which are currently limited by dim fluorophores and short trajectories. Future work will quantify the enhancement and further optimize the spectral overlap between the fluorophore and the substrate. An assessment of the biocompatibility of these and other plasmonic substrates would also help direct future efforts in this field. Using an objective heater, bacteria on the patterned gold surfaces could be monitored over several hours for signs of cell division. Alternatively, one could attempt to culture bacteria from a portion of a sample after imaging, as in Deich et al., 2004.

Experiments pairing plasmonic substrates with bacteria besides *V. cholerae* would also be of use, if only for a better understanding of potential side-effects of the cells coming in contact with the nanoparticle surface. *V. cholerae* is Gram-negative (Matson et al., 2007), and it is possible that the nanostructured substrates will be more toxic to Gram-positive bacteria. Cells with different envelope thicknesses would also experience enhancement to varying degrees. These plasmonic substrates could also be used for imaging proteins associated with the outer membrane, or free proteins inside the periplasm. In short, there are many potential applications for this plasmon-enhanced protocol.

5.3 Conclusions

Single-molecule imaging in live bacteria is a young and exciting field. As the field matures and opens up to more applications, direct observation of protein motion and interactions will reveal answers to biomedical questions. What we learn from the ToxR regulon may inform investigations of other membrane-bound transcription activation systems. Coupling such systems to noble metal nanostructured surfaces will allow us to push even further beyond the diffraction limit.

Appendix

This appendix contains source code for MATLAB programs developed for data analysis described in this dissertation.

Tracking

`Tracker`

`Tracker` is the function that examines super-resolution localization data and creates nearest-neighbor trajectories. (Version 4.3.1)

- **Input:** localization data
- **Output:** a pair of data files (with different sort orders) assigning each molecule to a trajectory

```
function output_sorted = Tracker(fit_file, input_param, out_folder)
versionNum = '4.3.1';
if nargin < 3
    %% Defaults if nargin < 3
    out_folder = fit_file;
end
%% Column Headers
roi_column = 1;
frame_column = 2;
x_column = 10;
y_column = 12;
last_original_col = 16;
frame_diff_col = 17;
disp_col = 18;
disp_alt_col = 19;
match_column = 20;
track_column = 21;
%% Prepare input & output
[movie_path, ~, ~] = fileparts(fit_file);
[~, movie_name, ~] = fileparts(movie_path);
movie_name = movie_name(1:strfind(movie_name, ' fits') - 1);
if ~exist(out_folder, 'dir')
    mkdir(out_folder)
end
%% Read input parameters
input_parameters = importdata(input_param, '\t', 1);
stationary = input_parameters.data(1);
max_dark_frames = input_parameters.data(2);
```

```
int_time = input_parameters.data(3);
px_size = input_parameters.data(4);
if stationary == 0 %Moving molecules
    max_disp_100 = input_parameters.data(5);
    max_disp_frame = input_parameters.data(6);
else %if stationary == 1 %Stationary molecules
    max_disp_abs = input_parameters.data(5);
    max_disp_nm = input_parameters.data(6);
end
clear input_parameters
%% Maximum difference in frames
max_frame_diff = max_dark_frames + 1;
%% Write parameter file
param_fit_file = [out_folder filesep 'track_param.dat'];
param_file = fopen(param_fit_file, 'w');
headers = ['Stationary' '\t' 'Dark Frames' '\t' 'Int. Time (ms)' '\t' ...
    'Pixel Size (nm)' '\t'];
params = [stationary max_dark_frames int_time px_size];
if stationary == 1 %Moving molecules
    fprintf(param_file, [headers 'Max Disp. (px)' '\t' 'Max Disp. (nm)' ...
        '\n']);
    fprintf(param_file, ['%g' repmat(['\t' '%g'], 1, 5) '\n'], params', ...
        max_disp_abs, max_disp_nm);
    param_msg = sprintf(['Num dark frames: %d' '\n' 'Max disp. (px): ' ...
        '%2.2f' '\n' '          (nm): %.4g' '\n' 'Integration time ' ...
        '(ms): %d' '\n' 'Pixel size (nm): %d' '\n'], max_dark_frames, ...
        max_disp_abs, max_disp_nm, int_time, px_size);
else %if stationary == 0 %Stationary molecules
    fprintf(param_file, [headers 'Max Disp./100 ms (nm)' '\t' ...
        'Max Disp./Frame (px)' '\n']);
    fprintf(param_file, ['%g' repmat(['\t' '%g'], 1, 5) '\n'], params', ...
        max_disp_100, max_disp_frame);
    param_msg = sprintf(['Num dark frames: %d' '\n' 'Max disp. ' ...
        '(nm/100 ms): %2.2f' '\n' '          (px/frame): %.4g' '\n' ...
        'Integration time (ms): %d' '\n' 'Pixel size (nm): %d' '\n'], ...
        max_dark_frames, max_disp_100, max_disp_frame, int_time, px_size);
end
%% Display parameters
disp('Parameters used:')
disp(param_msg)
fclose(param_file);
%% Import data
if ~exist(fit_file, 'file')
    error('Could not find -allgood data file.')
end
spotlist_struc = importdata(fit_file, '\t', 1);
spots = spotlist_struc.data;
clear spotlist_struc.data
[last_row, ~] = size(spots);
spots(last_row, track_column) = 0; %Add columns of zeros
%% Prepare output files
output_name = [out_folder filesep movie_name ' tracks.dat'];
output_sorted = [out_folder filesep movie_name ' tracks_sort.dat'];
% Open output for writing
output_file = fopen(output_name, 'wt');
output_file2 = fopen(output_sorted, 'wt');
% Write output column headers
headers = ['Frame' '\t' 'Molecule' '\t' 'Amplitude' '\t' '+/-' '\t' ...
    'Offset' '\t' '+/-' '\t' 'Width' '\t' '+/-' '\t' 'X Center' '\t' ...
```

```
    '+/-' '\t' 'Y Center' '\t' '+/-' '\t' 'Good Fit?' '\t' 'Integral' ...
    '\t' 'Small box width' '\t'];
fprintf(output_file, ['ROI' '\t' headers 'Track #' '\n']);
fprintf(output_file2, ['ROI' '\t' 'Track #' '\t' headers '\n']);
%% Determine number of ROIs
num_rois = spots(last_row, roi_column); %the last ROI is in the last row
%% Start looking for points
% Prime track counter.
track_num = 0;
for r = 1:num_rois
    start_new_track = 1;
    roi_finished = 0;
    roi_msg = sprintf('Looking through ROI %d', r);
    disp(roi_msg)
    while roi_finished == 0
        curr_roi_rows = find(spots(:, roi_column) == r);
        %% Set up
        if start_new_track == 1
            % Find the first unmatched point
            curr_row = find(spots(:, roi_column) == r & ...
                spots(:, match_column) == 0, 1);
            track_num = track_num + 1;
            spots(curr_row, match_column) = 1; %Set as matched
            spots(curr_row, track_column) = track_num;
        end
        % Find the frame differences
        spots(curr_roi_rows, frame_diff_col) = spots(curr_roi_rows, ...
            frame_column) - spots(curr_row, frame_column);
        % Find the displacements
        spots(curr_roi_rows, disp_col) = sqrt((spots(curr_roi_rows, ...
            x_column) - spots(curr_row, x_column)).^2 + ...
            (spots(curr_roi_rows, y_column) - spots(curr_row, y_column)).^2);
        % Find the displacements per frame
        spots(curr_roi_rows, disp_alt_col) = spots(curr_roi_rows, disp_col)...
            ./abs(spots(curr_roi_rows, frame_diff_col));
        %% Find potential matches
        if stationary == 1 %Stationary molecules
            potential_matches = spots(...
                spots(:, roi_column) == r & ... %current roi
                spots(:, match_column) == 0 & ... %unmatched
                spots(:, frame_diff_col) > 0 ... %next frame comes after
                & spots(:, frame_diff_col) <= max_frame_diff & ... %within limit
                spots(:, disp_col) < max_disp_abs, :); %within limit
        else %if stationary == 0 %Moving molecules
            potential_matches = spots(...
                spots(:, roi_column) == r & ... %current roi
                spots(:, match_column) == 0 & ... %unmatched
                spots(:, frame_diff_col) > 0 & ... %next frame comes after
                spots(:, frame_diff_col) <= max_frame_diff & ... %within limit
                spots(:, disp_alt_col) < max_disp_frame, :); %within limit
        end
        num_matches = size(potential_matches, 1);
        if num_matches == 1
            best_match_row = find(ismember(spots, potential_matches, 'rows'));
            spots(best_match_row, track_column) = track_num; %set track number
            spots(best_match_row, match_column) = 1; %mark as matched
            curr_row = best_match_row; %adjust row to compare against
            start_new_track = 0;
        elseif num_matches > 1 %choose the nearest neighbor in space
```

```
            if stationary == 1 %Stationary molecules
                min_disp_row = find(potential_matches(:, disp_col) == ...
                    min(potential_matches(:, disp_col)), 1);
            else %if stationary == 0 %Moving molecules
                min_disp_row = find(potential_matches(:, disp_alt_col) == ...
                    min(potential_matches(:, disp_alt_col)), 1);
            end
            num_matches_2 = length(min_disp_row);
            if num_matches_2 == 1
                best_match_row = find(ismember(spots, ...
                    potential_matches(min_disp_row, :), 'rows'));
            elseif num_matches_2 > 1 %choose the nearest neighbor in time
                potential_matches_2 = potential_matches(min_disp_row, :);
                min_fdiff_row = find(potential_matches_2(:, ...
                    frame_diff_col) == min(potential_matches_2(:, ...
                    frame_diff_col)), 1);
                best_match_row = find(ismember(spots, ...
                    potential_matches(min_fdiff_row, :), 'rows'));
            else
                error('Something went wrong. Your matches disappeared.')
            end
            spots(best_match_row, track_column) = track_num; %set track number
            spots(best_match_row, match_column) = 1; %mark as matched
            curr_row = best_match_row; %adjust row to compare against
            start_new_track = 0;
        else %if num_matches == 0
            start_new_track = 1;
        end
        num_unmatched_rows = histc(spots(curr_roi_rows, match_column), 0);
        matched = (size(curr_roi_rows, 1) - num_unmatched_rows);
        percentDone(matched, size(curr_roi_rows, 1))
        if num_unmatched_rows == 0
            roi_finished = 1;
        end
    end
end
%% Write output
out_format = ['%g' '\t' '%g' '\t' '%g' repmat(['\t' '%6.3f'], 1, 10) ...
    repmat(['\t' '%g'], 1, 3)];
sorted_rows = sortrows(spots, [roi_column track_column frame_column]);
sorted_spots = [sorted_rows(:, roi_column) sorted_rows(:, track_column) ...
    sorted_rows(:, frame_column:last_original_col)];
fprintf(output_file, [out_format '\t' '%g' '\n'], ...
    [spots(:, 1:last_original_col) spots(:, track_column)]');
fprintf(output_file2, ['%g' '\t' out_format '\n'], sorted_spots');
fprintf(output_file, ['Data from %s' '\n'], fit_file);
fprintf(output_file2, ['Data from %s' '\n'], fit_file);
fprintf(output_file, ['Tracks found using Tracker v.%d' '\n'], versionNum);
fprintf(output_file2, ['Tracks found using Tracker v.%d' '\n'], versionNum);
fclose all;
end
```

Mean-squared displacement analysis

MSD_calc

MSD_calc calculates the mean-squared displacement of tracks found using Tracker. Tracks may not use any dark frames. (Version 2.2.0)

- **Input:** a data file of trajectories (sorted) from Tracker and whether to allow blinking (dark frames)
- **Output:** a list of MSDs for each time lag and a file containing the parameters used

```
function output_file = MSD_calc(sorted_track_file, param_data, ID_num, ...
    output_path, output_name)
versionNum = '2.2.0';
%% Parameters
params = importdata(param_data, '\t', 1);
int_time = params.data(3); %ms
px_size = params.data(4); %nm
time_delay = 0; %ms
%% Column Definitions (from input)
ROI_column = 1;
track_column = 2;
frame_column = 3;
x_column = 11;
y_column = 13;
%% Conversions
px_um2 = (px_size / 1000).^2; % Convert from nm to um and square it;
time_delay_s = time_delay / 1000; % Convert from ms to s
int_time_s = int_time / 1000; % Convert from ms to s
%% Import track file
track_struc = importdata(sorted_track_file, '\t', 1);
track_data = track_struc.data;
%% Prep output file
if ~strcmp(output_path(end), filesep)
    output_path = [output_path filesep];
end
output_file = [output_path output_name '_MSD.dat'];
MSD_output = fopen(output_file, 'wt');
% First part of the output header
fprintf(MSD_output, ['Sample ID' '\t' 'ROI #' '\t' 'Track #' '\t' 'MSD Slope']);
%% Find the number of ROIs
last_ROI = max(track_data(:, ROI_column));
long_frames = 0;
%% Find the maximum number of tau values
for rr = 1:last_ROI
    %% Find rows for a given ROI
    ROI_start_row = find(track_data(:, ROI_column) == rr, 1);
    ROI_end_row = find(track_data(:, ROI_column) == rr, 1, 'last');
    last_track = max(track_data(ROI_start_row:ROI_end_row, track_column));
    for tt = 1:last_track
        num_frames = sum(track_data(ROI_start_row:ROI_end_row, ...
            track_column) == tt);
        if num_frames > long_frames
            long_frames = num_frames;
        end
```

```
    end
end
max_tau = (2/3)*(long_frames);
%% Second part of the output header
for j = 1:ceil(max_tau)
    curr_tau = j * (time_delay_s + int_time_s);
    fprintf(MSD_output, ['\t' '%g'], curr_tau);
end
fprintf(MSD_output, '\n');
%% Prep matrix for holding MSD data
last_tau_column = 4 + ceil(max_tau);
all_MSD_data = zeros(10000, last_tau_column);
curr_row = 1;
%% Calculate MSDs
for r = 1:last_ROI
    %% Find rows for a given ROI
    ROI_start_row = find(track_data(:, ROI_column) == r, 1);
    ROI_end_row = find(track_data(:, ROI_column) == r, 1, 'last');
    last_track = max(track_data(ROI_start_row:ROI_end_row, track_column));
    for t = 1:last_track
        %% Find first & last frames in a given track (for a given ROI)
        tr_start = find(track_data(ROI_start_row:ROI_end_row, ...
            track_column) == t, 1);
        tr_end =  find(track_data(ROI_start_row:ROI_end_row, ...
            track_column) == t, 1, 'last');
        track_start_row = ROI_start_row + tr_start - 1;
        track_end_row = ROI_start_row + tr_end - 1;
        last_frame = max(track_data(track_start_row:track_end_row, ...
            frame_column));
        first_frame = min(track_data(track_start_row:track_end_row, ...
            frame_column));
        if (last_frame - first_frame) ~= 0
            %% Start step size loop
            all_step_data = NaN(1, ceil(max_tau));
            for step_size = 1:round((2/3) * (last_frame - first_frame));
                num_points = 0;
                sum_sq_disps = 0;
                for i = track_start_row:(track_end_row - step_size)
                    %% Calculate displacements
                    xdisp = track_data(i + step_size, x_column) - ...
                        track_data(i, x_column);
                    ydisp = track_data(i + step_size, y_column) - ...
                        track_data(i, y_column);
                    %% Square displacements
                    % and convert from nm to um
                    xdisp_um2 = (xdisp.^2 * px_um2);
                    ydisp_um2 = (ydisp.^2 * px_um2);
                    %% Update sum & number of points
                    sum_sq_disps = sum_sq_disps + xdisp_um2 + ydisp_um2;
                    num_points = num_points + 1;
                end
                msd = sum_sq_disps / num_points;
                all_step_data(1, step_size) = msd;
            end
            first_msd = all_step_data(1);
            max_msd = max(all_step_data);
            max_cols = find(all_step_data == max_msd);
            max_msd_tau = max_cols(1) * (time_delay_s + int_time_s);
            first_msd_tau = 1 * (time_delay_s + int_time_s);
```

```
            MSD_slope = (max_msd - first_msd)/(max_msd_tau - ...
                first_msd_tau);
            all_MSD_data(curr_row, 1:4) = [ID_num r t MSD_slope];
            all_MSD_data(curr_row, 5:last_tau_column) = all_step_data;
            curr_row = curr_row + 1;
        end
    end
end
%% Trim the MSD matrix
% Start from the first row that is all zeros (the track number will be 0).
first_zero_row = find((all_MSD_data(:, 3) == 0), 1);
all_MSD_data(first_zero_row:10000,:) = [];
%% Print the output
output_format = repmat('\t%g', 1, last_tau_column-1);
fprintf(MSD_output, ['%15.0f', output_format, '\n'], all_MSD_data');
fprintf(MSD_output, ['Data from %s' '\n'], sorted_track_file);
fprintf(MSD_output, ['Calculated using MSD_calc v.%d' '\n'], versionNum);
fclose all;
end
```

Cumulative probability distribution analysis

CPD_disps

CPD_disps takes the trajectory information from Tracker and calculates the displacements at each time lag. This is useful for calculating MSD and CPD curves. (Version 1.2.3)

- **Input:** a data file of trajectories from Tracker, parameters used for tracking
- **Output:** a file of displacements at each given time lag

```
function output_file = CPD_disps(sorted_track_file, param_data, ID_num, ...
    output_path, output_name)
versionNum = '1.2.3';
%% Parameters
params = importdata(param_data, '\t', 1);
int_time = params.data(3); %ms
px_size = params.data(4); %nm
time_delay = 0; %ms
%% ID number check
if ~isnumeric(ID_num)
    ID_num = str2double(ID_num);
end
%% Column Definitions (from input)
ROI_column = 1;
track_column = 2;
frame_column = 3;
x_column = 11;
y_column = 13;
%% Conversions
px_um2 = (px_size / 1000).^2; % Convert from nm to um and square it;
time_delay_s = time_delay / 1000; % Convert from ms to s
int_time_s = int_time / 1000; % Convert from ms to s
%% Import track file
track_struc = importdata(sorted_track_file, '\t', 1);
track_data = track_struc.data;
%% Prep output file
if ~strcmp(output_path(end), filesep)
    output_path = [output_path filesep];
end
output_file = [output_path output_name '_CPD-disps.dat'];
disp_output = fopen(output_file, 'wt');
% First part of the output header
fprintf(disp_output, ['Sample ID' '\t' 'ROI #' '\t' 'Track #' '\t' 'Tau']);
%% Find the number of ROIs
last_ROI = max(track_data(:, ROI_column));
long_frames = 0;
%% Find the maximum number of tau values
for rr = 1:last_ROI
    %% Find rows for a given ROI
    ROI_start_row = find(track_data(:, ROI_column) == rr, 1);
    ROI_end_row = find(track_data(:, ROI_column) == rr, 1, 'last');
    last_track = max(track_data(ROI_start_row:ROI_end_row, track_column));
    for tt = 1:last_track
        num_frames = sum(track_data(ROI_start_row:ROI_end_row, ...
```

```
            track_column) == tt);
        if num_frames > long_frames
            long_frames = num_frames;
        end
    end
end
max_tau = (2/3)*(long_frames);
%% Second part of the output header
for j = 1:(long_frames-1)
    curr_tau = j * (time_delay_s + int_time_s);
    fprintf(disp_output, ['\t' '%g'], curr_tau);
end
fprintf(disp_output, '\n');
%% Prep matrix for holding displacements
% last_tau_column = 4 + ceil(max_tau);
all_disp_data = zeros(10000, (long_frames - 1 + 3));
last_filled = 0;
%% Calculate MSDs
for r = 1:last_ROI
    %% Find rows for a given ROI
    ROI_start_row = find(track_data(:, ROI_column) == r, 1);
    ROI_end_row = find(track_data(:, ROI_column) == r, 1, 'last');
    last_track = max(track_data(ROI_start_row:ROI_end_row, track_column));
    for t = 1:last_track
        %% Find first & last frames in a given track (for a given ROI)
        tr_start = find(track_data(ROI_start_row:ROI_end_row, ...
            track_column) == t, 1);
        tr_end =  find(track_data(ROI_start_row:ROI_end_row, ...
            track_column) == t, 1, 'last');
        track_start_row = ROI_start_row + tr_start - 1;
        track_end_row = ROI_start_row + tr_end - 1;
        last_frame = max(track_data(track_start_row:track_end_row, ...
            frame_column));
        first_frame = min(track_data(track_start_row:track_end_row, ...
            frame_column));
        if (last_frame - first_frame) ~= 0
            %% Start step size loop
            % ('step_size' being the difference in frames)
            disp_data = NaN(ceil(max_tau), long_frames);
            for step_size = 1:round((2/3) * (last_frame - first_frame));
                k = 1;
                for i = track_start_row:(track_end_row - step_size)
                    k = k + 1;
                    %% Calculate displacement
                    xdiff = track_data(i + step_size, x_column) - ...
                        track_data(i, x_column);
                    ydiff = track_data(i + step_size, y_column) - ...
                        track_data(i, y_column);
                    curr_disp = sqrt((xdiff^2 * px_um2) + (ydiff^2 * px_um2));
                    % Store displacement
                    disp_data(step_size, 1) = step_size * (time_delay_s + int_time_s);
                    disp_data(step_size, k) = curr_disp;
                end
            end
            num_step_rows = find(isnan(disp_data(:, 1)), 1, 'first') - 1;
            numeric_disp_data = disp_data(1:num_step_rows, :);
            for kk = (last_filled + 1):(last_filled + num_step_rows)
                all_disp_data(kk, 1:3) = [ID_num r t];
            end
```

```
            all_disp_data((last_filled + 1):(last_filled + num_step_rows), ...
                4:(long_frames + 3)) = numeric_disp_data;
            last_filled = last_filled + num_step_rows;
        end
    end
end
%% Trim the CPD matrix
% Start from the first row that is all zeros (the track number will be 0).
first_zero_row = find((all_disp_data(:, 3) == 0), 1);
all_disp_data(first_zero_row:10000,:) = [];
%% Print the output
output_format = repmat('\t%g', 1, (long_frames + 2));
fprintf(disp_output, ['%15.0f', output_format, '\n'], all_disp_data');
fprintf(disp_output, 'Data from %s', sorted_track_file);
fprintf(disp_output, ['Calculated using CPD_disps v.%d' '\n'], versionNum);
fclose all;
disp('Complete.')
end
```

dispThresh

`dispThresh` takes displacement data calculated for determining CPDs (with `CPD_disps`) and trims it to include only tracks with a specified minimum number of time-lag values. This is useful for calculating MSDs and CPDs. (Version 1.2.0)

- **Input:** data file from `CPD_disps`
- **Output:** copy of the input data file containing only values from tracks that are longer than the specified threshold

```
function dispThresh(folder, disps_file, outFolder, tauThresh)
versionNum = '1.2.0';
%% Prepare flags
flag1 = 0;
flag2 = 0;
flag3 = 0;
%% Defaults
if nargin < 4 %numeric input = threshold value
    if nargin == 0
        tauThresh = 6; %default threshold is min. 6 time-lags (10 frames)
    elseif isnumeric(folder)
        tauThresh = folder;
        flag1 = 1;
    elseif isnumeric(disps_file)
        tauThresh = disps_file;
        flag2 = 1;
    elseif isnumeric(outFolder)
        tauThresh = outFolder;
        flag3 = 1;
    else
        tauThresh = 6; %minimum number of time-lags required
    end
end
%% Choose input
if nargin == 0 || flag1 == 1; %No folder or file specified
    [disps_file, folder] = uigetfile({'*.dat', 'data files (*.dat)'}, ...
        'Choose a track list');
    if disps_file == 0
        disp('No file selected.')
        return
    end
elseif nargin == 1 || flag2 == 1; %Folder and file entered as single string
    [folder, file, ext] = fileparts(folder);
    disps_file = [file ext];
end
if ~strcmp(folder(end), filesep)
    folder = [folder filesep];
end
%% Choose output
if nargin < 3 || flag3 == 1 %No output folder specified
    outFolder = uigetdir('Choose an output folder');
    if outFolder == 0;
        disp('No folder chosen.')
        return
    end
end
```

```
[~, outName, ~] = fileparts(disps_file);
if ~strcmp(outFolder(end), filesep)
    outFolder = [outFolder filesep];
end
if ~strcmp(disps_file(end-3:end), '.dat')
    disps_file = [disps_file '.dat'];
end
%% Import disps.data
try
    disps = importdata([folder disps_file], '\t', 1);
catch err
    error('Could not import data. Check that the file is not open.')
end
%% Columns
sampleCol = 1;
roiCol = 2;
trackCol = 3;
tauCol = 4;
%% Find info for tracks with at least the minimum number of tau values
minTau = min(disps.data(:, tauCol));
threshVal = minTau*tauThresh;
overThresh = disps.data(abs(disps.data(:, tauCol) - threshVal) < 0.01, ...
    sampleCol:trackCol);
[nRows, ~] = size(overThresh);
[dataRows, dataCols] = size(disps.data);
matchList(dataRows, dataCols) = 0;
lastMatch = 0;
%% Collect all info for matching tracks
for iRow = 1:nRows
    matches = disps.data(...
          ismember(disps.data(:, sampleCol),overThresh(iRow, sampleCol)) ...
        & ismember(disps.data(:, roiCol),   overThresh(iRow, roiCol)) ...
        & ismember(disps.data(:, trackCol), overThresh(iRow, trackCol)), :);
    nMatches = size(matches, 1);
    matchList(lastMatch + 1: lastMatch + nMatches, :) = matches;
    lastMatch = lastMatch + nMatches;
end
%% Trim out extra blank rows
matchList = matchList(matchList(:, sampleCol) ~= 0, :);
%% Prepare output
outputFile = [outFolder filesep outName ' ' num2str(tauThresh) ' tau.dat'];
output = fopen(outputFile, 'wt');
%% Write output
headerFormat = ['%s' repmat(['\t' '%s'], 1, dataCols - 1) '\n'];
dataFormat = ['%15.0f' repmat(['\t' '%g'], 1, (dataCols - 1)) '\n'];
fprintf(output, headerFormat, disps.colheaders{:});
fprintf(output, dataFormat, matchList');
fprintf(output, ['Data from %s' '\n'], [folder disps_file]);
fprintf(output, ['File created with dispThresh, v.' '%s'], versionNum);
close all;
fclose all;
disp('Complete.')
end
```

CPD_calc

`CPD_calc` calculates the cumulative probability for each squared step size at each time-lag. (Version 1.3.0)

- **Input:** data file from `CPD_disps`
- **Output:** data file of squared step sizes, time-lags and their corresponding probabilities

```
function CPD_calc(folder, filename, outFolder)
versionNum = '1.3.0';
%% Define Columns
% roi_column = 2;
% track_column = 3;
tau_column = 4;
disp2_column = 5;
%% Input
if nargin == 0
    [filename, folder] = uigetfile({'*.dat', 'Displacment data (*.dat)'}, ...
        'Choose a displacement data file');
else
    if ~strcmp(folder(end), filesep)
        folder = [folder filesep];
    end
end
if nargin < 3
    outFolder = pwd;
end
if ~strcmp(outFolder(end), filesep)
    outFolder = [outFolder filesep];
end
cpd_data = importdata([folder filename], '\t', 1);
if ~iscell(cpd_data)
    max_tau = max(cpd_data.data(:, tau_column));
    min_tau = min(cpd_data.data(:, tau_column));
else
    msg1 = sprintf('No data in %s. Moving on.', filename);
    disp(msg1)
    return
end
%% Setup
blank_rows = 5000;
all_cpd_data = zeros(blank_rows, 3);
last_filled_row = 0;
%% Arrange Plot
figure('Color', 'w');
set(gca, 'ColorOrder', jet);
set(gca, 'YScale', 'log', 'YLim', [0 1])
hold all;
cpd_data.data(cpd_data.data(:, disp2_column:end) == 0) = NaN;
tolerance = 0.0001;
for tau = min_tau:min_tau:max_tau
    curr_tau_rows = abs(tau - cpd_data.data(:, tau_column)) < tolerance;
    disps = reshape(cpd_data.data(curr_tau_rows, disp2_column:end), [], 1);
    disps(disps == 0) = NaN;
    disps = disps(~isnan(disps));
```

```
    uniq_disps = reshape(unique(disps), [], 1);
    freq_disps = histc(disps, uniq_disps);
    num_disps = length(uniq_disps);
    cpd_plot_data = [uniq_disps freq_disps nan(num_disps, 3)];
    dispCount = sum(cpd_plot_data(:, 2));
    cpd_plot_data(1, 3) = cpd_plot_data(1, 2)/dispCount;
    for iRow = 2:num_disps
        cpd_plot_data(iRow, 3) = cpd_plot_data(iRow - 1, 3) + ...
            cpd_plot_data(iRow, 2)/dispCount;
    end
    cpd_plot_data(:, 4) = cpd_plot_data(:, 1).^2;
    r_sq = cpd_plot_data(:, 4);
    prob = cpd_plot_data(:, 3);
    tau_fill = tau*ones(num_disps, 1);
    plot(r_sq, prob);
    all_cpd_data((last_filled_row + 1):(last_filled_row + num_disps), :) ...
        = [tau_fill r_sq prob];
    last_filled_row = last_filled_row + num_disps;
end
set(gca, 'XScale', 'log', 'YScale', 'lin', 'XLim', [1e-6 1], 'YLim', [0 1]);
xlabel('squared step size (um^2)')
ylabel('P(r^2, tau)')
first_zero_row = find(all_cpd_data(:, 1) == 0, 1);
all_cpd_data(first_zero_row:blank_rows, :) = [];
output = fopen([outFolder strrep(filename, '.dat', '') '_CPD.dat'], 'w');
fprintf(output, ['tau' '\t' 'r^2 (um^2)' '\t' 'P' '\n']);
fprintf(output, ['%g' '\t' '%g' '\t' '%g' '\n'], all_cpd_data');
fprintf(output, ['Data from %s' '\n'], [folder filename]);
fprintf(output, 'Created with CPD_calc version %s\n', versionNum);
hold off;
saveas(gcf, [outFolder strrep(filename, '.dat', '') ' CPD plot'], 'fig');
saveas(gcf, [outFolder strrep(filename, '.dat', '') ' CPD plot'], 'jpg');
fclose all;
disp('Done.')
end
```

cpdParams

`cpdParams` creates the matrix of initial parameters used in `cpdFitting`. (Version 1.1.0)

- **Input:** parameters entered in reply to prompts
- **Output:** file of initial fitting parameters for `cpdFitting`

```
function initialParams = cpdParams
finished = 0;
while finished ~= 1
    %% Determine type of model
    numTerms = input('How many terms in the model (including immobile)? ');
    if numTerms == 0
        initialParams = 0;
        finished = 1;
    else
        if mod(numTerms, 1) ~= 0
            error('Input must be an integer.');
        end
        initialParams = zeros(1, 2*numTerms);
        immobileTerm = input(['Is there an immobile term?' '\n' '[1] Yes.' ...
            '\n' '(0) No.' '\n']);
        if isempty(immobileTerm)
            immobileTerm = 1;
        elseif sum(ismember([0 1], immobileTerm)) == 0 %if it's not 1 or 0
            disp('There is either one immobile term, or none.')
            immobileTerm = 1;
            disp('Using 1 immobile term.')
        end
        fprintf('\n')
        %% Enter parameter values
        for term = 1:numTerms - 1
            currFrac = sprintf(['What is the starting fraction for mode ' ...
                '%d? '], term);
            initialParams(1, term) = input(currFrac);
            currMSD = sprintf('What is the starting MSD for this mode? ');
            initialParams(1, numTerms + term) = input(currMSD);
        end
        initialParams(1, numTerms) = 0;
        if immobileTerm == 1
            initialParams(1, 2*numTerms) = 0;
        else
            initialParams(1, 2*numTerms) = input(['What is the starting ' ...
                'MSD for the last mode? ']);
        end
        stillSwapping = 1;
        while stillSwapping == 1
            %% Check for reasonableness of fractions
            currTotal = sum(initialParams(1, 1:numTerms - 1));
            while (currTotal - 1) > -0.001
                % The sum must be less than one so that the final term has some
                % contribution
                fprintf('\n')
                disp('The fractional contributions must add up to one.')
                msg1 = sprintf(['The current fractions add up to %2.3f ' ...
                    'and do not leave room for the final term.'], ...
                    currTotal);
```

```
    disp(msg1)
    swapTerm = input(['For which term would you like to ' ...
        'change the fraction? ']);
    if mod(swapTerm, 1) == 0 && swapTerm < numTerms - 1 && swapTerm > 0
        swapMsg = sprintf(['What is the starting fraction ' ...
            'for mode %d? '], swapTerm);
        swapFrac = input(swapMsg);
        initialParams(1, swapTerm) = swapFrac;
    elseif swapTerm > numTerms - 1 || swapTerm < 0
        msg2 = sprintf(['There are only %d terms, and the ' ...
            'last one is determined automatically.'], numTerms);
        disp(msg2)
    else
        disp('I don"t understand. Try again.')
    end
    currTotal = sum(initialParams(1, 1:numTerms));
end
clc
%% Display entered parameters
for nTerm = 1:numTerms - 1
    fprintf(['Fraction from mode %d:' '\t' '%2.3f' '\n'], nTerm, ...
        initialParams(1, nTerm));
end
lastFrac = 1 - currTotal;
fprintf(['Fraction from mode %d:' '\t' '%2.3f' '\n'], numTerms, ...
    lastFrac);
for iTerm = 1: numTerms
    fprintf(['MSD for mode %d:' '\t' '%4.4f um^s'], iTerm, ...
        initialParams(1, numTerms + iTerm));
    if initialParams(1, numTerms + iTerm) == 0
        fprintf(' (stationary)')
    end
    fprintf('\n')
end
fprintf('\n')
%% Confirm parameters
finished = input(['Does this set of parameters look okay to you?' ...
    '\n' '[1] Yes.' '\n' '(0) No.' '\n']);
if isempty(finished)
    finished = 1; %break out of while loop and finish program
elseif finished ~= 1
    fprintf('\n')
    startOver = input(['Do you want to' '\n' '[1] start over, or' ...
        '\n' '(0) change a value?' '\n']);
    if isempty(startOver)
        startOver = 1;
    end
end
if finished == 1
    stillSwapping = 0;
    startOver = 2;
end
if startOver == 0
    swapPart = input(['In which term do you want to change a ' ...
        'parameter? ']);
    if swapPart < numTerms || swapPart > 0
        swapPiece = input(['Do you want to change the ' '\n' ...
            '(1) fraction, or' '\n' '(2) MSD value?' '\n']);
        if swapPart == numTerms
```

```
                        if immobileTerm == 1
                            disp('That is determined automatically.')
                            disp(['If you want to change the type of ' ...
                                'model, you will need to start over.']);
                        elseif swapPiece == 1
                            disp('That is determined automatically.')
                        else
                            initialParams(1, numTerms + swapPart) = ...
                                input(['What is the starting MSD for the ' ...
                                'last mode? ']);
                        end
                    elseif swapPiece == 1
                        swapFrac2 = sprintf(['What is the starting ' ...
                            'fraction for mode %d? '], swapPart);
                        initialParams(1, swapPart) = input(swapFrac2);
                    elseif swapPiece == 2
                        swapFrac2 = sprintf(['What is the starting ' ...
                            'fraction for mode %d? '], swapPart);
                        initialParams(1, numTerms + swapPart) = input(swapFrac2);
                    else
                        disp('I don"t understand. Try again.')
                    end
                elseif swapPart > numTerms || swapPart <0
                    msg3 = sprintf('There are only %d terms.', numTerms);
                    disp(msg3)
                else
                    disp('I don"t understand. Try again.')
                end
                fprintf('\n')
            else %if startOver == 1
                clc
                stillSwapping = 0; %break out of while loop; go back to start
            end
        end
    end
end
if numTerms > 1
    disp('Parameters set.')
end
end
```

cpdFitting

`cpdFitting` fits a CPD curve to a specified model. (Version 2.0.1; based on `CPD` by Yi Liao)

- **Input:** a data file of displacements from `CPD_disps`, the model to fit with, and the localization precision to use
- **Output:** best fit parameters for the given data and model

```
function cpdFitting(model, sigmaValue, CPDfolder, filename, outFolder)
versionNum = '2.0.1';
%% Options
useColors = 0;
%% Fit Parameters
% The order of parameters is:
%
% * [alpha beta gamma MSD1 MSD2 MSD3]
%
% Set the first parameter (fraction) of your last term equal to zero, as
% well as the second parameter (MSD) of an immobile term. (Immobile term
% must come last.) For example,
%
% * three mobile terms: [alpha beta 0 MSD1 MSD2 MSD3]
% * two mobile, one immobile: [alpha beta 0 MSD1 MSD2 0]
% * one (mobile) term: [0 MSD1]
%
if nargin > 0
    switch model
        case '1-0'
            initialParams = [0 0.01];
        case '1-1'
            initialParams = [0.5 0 0.01 0];
        case '2-0'
            initialParams = [0.5 0 0.02 0.01];
        case '2-1'
            initialParams = [0.3 0.3 0 0.02 0.01 0];
        case '3-0'
            initialParams = [0.3 0.3 0 0.02 0.01 0.001];
        case '3-1'
            initialParams = [0.3 0.3 0.3 0 0.02 0.01 0.001 0];
        otherwise
            initialParams = cpdParams; %generate a set of parameters
    end
else %no inputs
    initialParams = cpdParams;
end
if initialParams == 0
    return %quit
end
lowerBound = zeros(1, length(initialParams));
upperBound = ones(1, length(initialParams));
options = optimset('Display', 'off'); %Don't show the lsqcurvefit status.
%% Input
if nargin <= 2
    if nargin <= 1
        sigmaValue = input('What sigma value (in nm) do you want to use? ');
```

```
    end
    [filename, CPDfolder] = uigetfile({'*.dat', 'CPD data (*.dat)'}, ...
        'Choose a CPD data file', pwd);
    if isequal(filename, 0)
        return
    end
end
if ~strcmp(CPDfolder(end), filesep)
    CPDfolder = [CPDfolder filesep];
end
cpd = importdata([CPDfolder filename], '\t', 1);
% Remove zeros (they're in as placeholders)
initialParams = initialParams(initialParams ~= 0);
lowerBound = lowerBound(initialParams ~=0);
upperBound = upperBound(initialParams ~= 0);
numParams = length(initialParams);
sampleName = folderFinder(CPDfolder, 2, 'end');
movieNameEnd = strfind(filename, '_CPD-disps');
movieName = filename(1:movieNameEnd - 1);
runTime = datestr(now, 'HHMM');
if nargin < 3
    outFolder = [pwd filesep sampleName filesep movieName ...
        ' CPD fits - ' runTime];
    if exist(outFolder, 'dir')
        outFolder = [outFolder datestr(now, 'SS')];
    end
    try
        mkdir(outFolder)
    catch err1
        outFolder = uigetdir(pwd, 'Choose an output folder.');
    end
end
if ~strcmp(outFolder(end), filesep)
    outFolder = [outFolder filesep];
end
tauValues = cpd.data(:, 1);
minTau = min(tauValues); %s
tolerance = 0.0001; %for finding steps and P values at a given tau
numTaus = 4; %just the first 4
masterList(numTaus, numParams + 2) = 0;
f = figure;
set(f, 'Color', 'w')
hold on
numImmobile = ~mod(numParams, 2);
numMobile = floor(numParams/2) + 1 - numImmobile;
titleText = sprintf('%d Mobile, %d Immobile, s = %d', numMobile, ...
    numImmobile, sigmaValue);
fitParams = [initialParams 1/(numMobile + numImmobile)];
fitLow = [lowerBound 0];
fitUp = [upperBound 1];
color = {'b', 'g', 'r', 'c', 'm', 'y', 'k'};
for currTau = 1:numTaus
    currTauValue = currTau*minTau; %s
    squaredSteps = cpd.data(abs(cpd.data(:, 1) - currTauValue) < ...
        tolerance, 2); %um^2
    PValues = cpd.data(abs(cpd.data(:, 1) - currTauValue) < ...
        tolerance, 3); %P(r^2, tau)
    if length(squaredSteps) > 2
        %% Fit CPD curve
```

```
        [resultParams, ~, residual] = lsqcurvefit(@cpdFitFn, fitParams, ...
            squaredSteps, PValues, fitLow, fitUp, options);
        reducedChiSq=sum(residual.^2)/(length(residual) - numParams);
        resultingPoints = cpdFitFn(resultParams, squaredSteps);
        hold all
        subplot(3, 3, 1:6); %CPD plot
        if useColors == 1
            semilogx(squaredSteps, PValues, [color{currTau} '-'])
        else
            semilogx(squaredSteps, PValues, 'r-')
        end
        hold on
        if useColors == 1
            semilogx(squaredSteps, resultingPoints, [color{currTau} '--'])
        else
            semilogx(squaredSteps, resultingPoints, 'b-')
        end
        xLim = [1e-6 1];
        set(gca, 'XLim', xLim, 'YLim', [1e-4 1])
        ylabel('1 - P(r^{2}, \tau)')
        title(titleText)
        hold on
        subplot(3, 3, 7:9); %residual plot
        if useColors == 1
            semilogx(squaredSteps, residual, [color{currTau} '-'])
        else
            semilogx(squaredSteps, residual, 'b-')
        end
        set(gca, 'XLim', xLim, 'YLim', [-0.05 0.05])
        ylabel('Residual')
        xlabel('Squared step size ({\mu}m^2)')
        resultParams1 = resultParams(1:end - 1);
        masterList(currTau, :) = [currTauValue resultParams1 reducedChiSq];
    else
        masterList(currTau, :) = [currTauValue zeros(1, numParams + 1)];
    end
end
nTerms = numMobile + numImmobile;
if nTerms > 1
    masterList(:, end+1) = 1 - sum(masterList(:, 2:nTerms), 2);
        %because you start in column 2, the column you end in is the same # as
        %the number of terms (i.e. 2 + nTerms-1 - 1 = nTerms)
end
%% Means & Std Devs
stats(2, size(masterList, 2) - 1) = 0; %time-lags are in first column
stats(1, :) = mean(masterList(1:4, 2:end));
stats(2, :) = std(masterList(1:4, 2:end));
%% Prepare output
fitType = [' m' num2str(numMobile) 'i' num2str(numImmobile)];
sig = [' s' num2str(sigmaValue)];
% Output folder
outName = [outFolder movieName fitType sig ' CPD'];
% Figure
saveas(f, [outName 's'], 'fig');
saveas(f, [outName 's'], 'jpg');
% Data file
outputName = [outName '_fit.dat'];
outputFile = fopen(outputName, 'wt');
% Header
```

```
nParams = 1:numParams;
paramNums = strcat({'Param '}, reshape(cellstr(num2str(nParams(:))), ...
    size(nParams)));
outputHeader = ['Tau (s)' paramNums(:)' 'Reduced Chi^2'];
if nTerms > 1
    outputHeader(end+1) = {['Param ' num2str(numParams + 1)]};
end
nRepCols = size(outputHeader, 2) - 1;
% Formats
headerFormat = ['%s' repmat(['\t' '%s'], 1, nRepCols) '\n'];
paramFormat = ['%s' repmat(['\t' '%g'], 1, nRepCols) '\n'];
outputFormat = ['%g' repmat(['\t' '%g'], 1, nRepCols) '\n'];
statsFormat = ['%s' repmat(['\t' '%g'], 1, nRepCols) '\n'];
footerFormat = ['%s' '\n'];
paramDetails = ['initial values' num2cell(initialParams); ...
                'lower bound' num2cell(lowerBound); ...
                'upper bound' num2cell(upperBound)];
paramDetails(4, :) = ['sigma (nm)' num2cell(sigmaValue) ...
    num2cell(nan(size(upperBound(2:end))))];
if nTerms > 1
    paramDetails(:, end + 1) = num2cell(nan(4, 1));
end
paramDetails(:, end + 1) = num2cell(nan(4, 1));
statDetails(1:2, 1) = {'Mean'; 'Std. Dev'};
statDetails(:, 2:size(masterList, 2)) = num2cell(stats);
    %columns in masterList - 1 for taus + 1 because we're starting in col 2
footer1 = sprintf('Data from %s', [CPDfolder filename]);
footer2 = sprintf('Calculated using cpdFitting %s', versionNum);
footerDetails = {footer1; footer2};
paramDetails2 = paramDetails';
statDetails2 = statDetails';
%% Print output
fprintf(outputFile, headerFormat, outputHeader{:});
fprintf(outputFile, paramFormat, paramDetails2{:});
fprintf(outputFile, outputFormat, masterList');
fprintf(outputFile, outputFormat, nan(1, nRepCols + 1)');
fprintf(outputFile, statsFormat, statDetails2{:});
fprintf(outputFile, footerFormat, footerDetails{:});
%% CPD Fitting Function
    function cpdFit = cpdFitFn(parameters, rSquared)
        % cpdFit version 1.0.0
        % Based on |choose_function| by Yi Liao.
        numTerms = floor((length(parameters) - 1)/2) + 1; % 2 params/term
        sigmaSquared = (sigmaValue/1000)^2; % Convert nm to um^2
        fn = 0;
        usedFraction = 0;
        if numTerms == 1
            fn = exp(-rSquared/(parameters(1) + sigmaSquared));
            usedFraction = usedFraction + parameters(2); %should be 1
        else
            for currTerm = 1:numTerms - 1
                fn = fn + parameters(currTerm)*exp(-rSquared/...
                    (parameters(numTerms - 1 + currTerm) + sigmaSquared));
                usedFraction = usedFraction + parameters(currTerm);
            end
        end
        % Last term
        if mod(length(parameters) - 1, 2) == 0
            finalMsd = 0; %immobile
```

```
        else
            finalMsd = parameters(end - 1); %mobile
        end
        if length(parameters) - 1 > 1 %Don't add anything if it's just 1 mobile term
            parameters(end) = 1 - usedFraction;
            fn = fn + parameters(end)*exp(-rSquared/(finalMsd + sigmaSquared));
        end
        cpdFit = 1 - fn; %Fits to P(r^2, tau), i.e. 1 - C(r^2, tau)
    end
disp('Complete.')
fclose all;
end
```

bootstrapCPD

`bootstrapCPD` samples the displacement data generated by `CPD_disps` and fits the resulting CPD curves to determine the error in the fits and parameters. (Version 1.0.0)

- **Input:** a data file of displacements, names for each strain, how many samples to take of the dataset, initial fit parameters for the CPD curve
- **Output:** a `*.mat` file containing the sampled data, the best fit curves, and the fit parameters, and a set of figures

```
function outputFile = bootstrapCPD(nSamplesOrPercent, initialParams, ...
    dataFile, strainTitle, strainName)
%% Parameters
% Raw Data
intTime = 0.05; %seconds
tolerance = intTime/5;
% Fitting
% The model is 2 mobile terms and 1 immobile term
sigmaValue = 30; %nm
nTauVals = 4;
% Sampling
nPopulations = 100;
%% Options
usePercents = 1; %Determine sampling from percent of total displacements
plotting = 1; %Plot the bootstrapped CPD curves and fits
if nargin >= 1
    if nSamplesOrPercent <= 1
        fracData = nSamplesOrPercent;
        useFractions = 1;
    else
        nSamples = nSamplesOrPercent;
        useFractions = 0;
    end
else %no input arguments
    useFractions = usePercents;
    fracData = 1.0; %s %if usePercents == 1
    nSamples = 2000;    %if usePercents == 0
    initialParams = [0.3 0.3 0 0.02 0.01 0];
end
% Font size
titleFontSize = 16; %pt
labelFontSize = 12; %pt
%% Input
if nargin < 5
    [dataFilename, dataFolder] = uigetfile('.dat', 'Choose a disps data file');
    dataFile = fullfile(dataFolder, dataFilename);
    strainTitle = input('Strain name (for plotting): ', 's');
    strainName = input('Strain name (for *.mat file): ', 's');
else
    [dataFolder, ~, ~] = fileparts(dataFile);
end
if iscell(dataFile)
    nStrains = max(size(dataFile));
else
    nStrains = 1;
end
```

```
%% Output path
outFolder = dataFolder;
%% CPD Model
lowerBound = zeros(1, length(initialParams));
upperBound = ones(1, length(initialParams));
options = optimset('Display', 'off'); %Don't show the lsqcurvefit status.
% Remove zeros (they're in as placeholders)
initialParams = initialParams(initialParams ~= 0);
lowerBound = lowerBound(initialParams ~=0);
upperBound = upperBound(initialParams ~= 0);
numParams = length(initialParams);
numImmobile = ~mod(numParams, 2);
numMobile = floor(numParams/2) + 1 - numImmobile;
fitParams = [initialParams 1/(numMobile + numImmobile)];
fitLow = [lowerBound 0];
fitUp = [upperBound 1];
frameTime = intTime*1000; %ms
%% State parameters used
disp('Bootstrapping CPDs')
if useFractions == 1
    msgTxt = sprintf('Sampling %d%% of each dataset %d times', ...
        fracData*100, nPopulations);
else %useFractions == 0
    msgTxt = sprintf('Sampling %d datapoints %d times', nSamples, ...
        nPopulations);
end
disp(msgTxt)
disp('Fitting')
colorOrder = {'b' 'r' [0 0.7 0] 'k'};
strains = struct;
for iStrain = 1:nStrains
    fprintf('\n')
    if nStrains > 1
        inputData = importdata(dataFile{iStrain}, '\t', 1);
        currStrainTitle = strainTitle{iStrain};
        currStrainName = strainName{iStrain};
    else
        inputData = importdata(dataFile, '\t', 1);
        currStrainTitle = strainTitle;
        currStrainName = strainName;
    end
    strains(iStrain).name = currStrainName;
    disp(strains(iStrain).name)
    strains(iStrain).inputs = initialParams;
    strains(iStrain).lowBounds = lowerBound;
    strains(iStrain).upBounds = upperBound;
    disp('Generating CPD curve from full dataset')
    % overall = struct;
    for iTauVal = 1:nTauVals
        strains(iStrain).overall(iTauVal).name = sprintf(['tau' '%d'], iTauVal);
        strains(iStrain).overall(iTauVal).headers = inputData.textdata;
        strains(iStrain).overall(iTauVal).allcols = ...
            inputData.data(inputData.data(:, 4) < ...
            (iTauVal*intTime + tolerance) & ...
            inputData.data(:, 4) > ((iTauVal - 1)*intTime + tolerance), :);
        strains(iStrain).overall(iTauVal).disp = reshape(...
            strains(iStrain).overall(iTauVal).allcols(:, 5:end), [], 1);
        strains(iStrain).overall(iTauVal).disp(...
            strains(iStrain).overall(iTauVal).disp == 0) = NaN;
```

```
    strains(iStrain).overall(iTauVal).disp = ...
        strains(iStrain).overall(iTauVal).disp(...
        ~isnan(strains(iStrain).overall(iTauVal).disp));
    strains(iStrain).overall(iTauVal).uniq = ...
        unique(strains(iStrain).overall(iTauVal).disp);
    strains(iStrain).overall(iTauVal).freq = ...
        histc(strains(iStrain).overall(iTauVal).disp, ...
        strains(iStrain).overall(iTauVal).uniq);
    strains(iStrain).overall(iTauVal).prob = ...
        cumsum(strains(iStrain).overall(iTauVal).freq)./...
        sum(strains(iStrain).overall(iTauVal).freq);
    strains(iStrain).overall(iTauVal).maxD = ...
        max(strains(iStrain).overall(iTauVal).disp);
    strains(iStrain).overall(iTauVal).minD = ...
        min(strains(iStrain).overall(iTauVal).disp);
end
maxima = [strains(iStrain).overall.maxD];
minima = [strains(iStrain).overall.minD];
strains(iStrain).figName = sprintf('%s - CPD of all data', currStrainTitle);
strains(iStrain).hF = figure('Color', 'w');
strains(iStrain).hAfits = subplot(3, 3, 1:6);
strains(iStrain).hAresids = subplot(3, 3, 7:9);
hold all
strains(iStrain).hPfits(8) = 0;
strains(iStrain).hPresids(8) = 0;
for iTau = 1:nTauVals
    tauMsg = sprintf('tau %d', iTau);
    disp(tauMsg)
    disp('Calculating fits')
    axes(strains(iStrain).hAfits) %#ok<LAXES>
    hold all
    % Plot CPD curve
    strains(iStrain).hPfits(iTau) = ...
        plot([strains(iStrain).overall(iTau).uniq].^2, ...
        [strains(iStrain).overall(iTau).prob], ...
        'Color', colorOrder{iTau});
    %% CPD fitting (overall)
    [resultParamListAll, ~, strains(iStrain).overall(iTau).residuals, ...
        ~, ~, ~, strains(iStrain).overall(iTau).Jacobian ] = ...
        lsqcurvefit(@cpdFitFn, fitParams, ...
        strains(iStrain).overall(iTau).uniq.^2, ...
        strains(iStrain).overall(iTau).prob, ...
        fitLow, fitUp, options);
    % Transpose parameter list to fit in output matrix
    strains(iStrain).overall(iTau).resultParams = resultParamListAll';
    strains(iStrain).overall(iTau).resultingPoints = ...
        cpdFitFn(strains(iStrain).overall(iTau).resultParams, ...
        strains(iStrain).overall(iTau).uniq.^2);
    strains(iStrain).overall(iTau).reducedChiSq = ...
        sum(strains(iStrain).overall(iTau).residuals.^2)/...
        (length(strains(iStrain).overall(iTau).residuals) - numParams);
    strains(iStrain).overall(iTau).resultingPoints = ...
        cpdFitFn(strains(iStrain).overall(iTau).resultParams, ...
        strains(iStrain).overall(iTau).uniq.^2);
    hold on
    % Plot CPD fit
    strains(iStrain).hPfits(iTau+4) = ...
        plot([strains(iStrain).overall(iTau).uniq].^2, ...
        [strains(iStrain).overall(iTau).resultingPoints], '--', ...
```

```
        'Color', [0.3 0.3 0.3]);
    hold all
    % Plot residuals
    axes(strains(iStrain).hAresids) %#ok<LAXES>
    strains(iStrain).hPresids(iTau+4) = ...
        plot([strains(iStrain).overall(iTau).uniq].^2, ...
        strains(iStrain).overall(iTau).residuals, ...
        'Color', colorOrder{iTau});
end
hold off
axes(strains(iStrain).hAfits) %#ok<LAXES>
set(strains(iStrain).hAfits, 'XLim', [1e-6 1], 'YLim', [0 1], ...
    'XScale', 'log', 'FontSize', labelFontSize)
legend(strains(iStrain).hPfits(1:5), ...
    {'50 ms' '100 ms' '150 ms' '200 ms' 'fits'}, 2)
title(strains(iStrain).figName, 'FontSize', titleFontSize)
ylabel('P(r^2, tau)', 'FontSize', labelFontSize)
axes(strains(iStrain).hAresids) %#ok<LAXES>
set(strains(iStrain).hAresids, 'XLim', [1e-6 1], ...
    'YLim', [-0.05 0.05], 'XScale', 'log', 'FontSize', labelFontSize)
xlabel('squared step size (um^2)', 'FontSize', labelFontSize)
ylabel('y_{fit} - y_{data}', 'FontSize', labelFontSize)
%% MSD from CPD fitting (overall)
strains(iStrain).hFmsd = figure('Color', 'w');
strains(iStrain).hPfast(nPopulations, nTauVals) = 0;
strains(iStrain).hPslow(nPopulations, nTauVals) = 0;
hold all
for ttt = 1:nTauVals
    % Fast
    strains(iStrain).hPfast(:, ttt) = plot(ttt*0.05, ...
        strains(iStrain).overall(ttt).resultParams(3, :), 'r.');
    % Slow
    strains(iStrain).hPslow(:, ttt) = plot(ttt*0.05, ...
        strains(iStrain).overall(ttt).resultParams(4, :), 'b.');
end
set(gca, 'XLim', [0 0.25])
xlabel('tau (s)', 'FontSize', labelFontSize)
ylabel('MSD (um^2)', 'FontSize', labelFontSize)
msdTitle = sprintf('%s - MSD vs tau from CPDs (overall)', ...
    currStrainTitle);
title(msdTitle, 'FontSize', titleFontSize)
legend([strains(iStrain).hPfast(1) strains(iStrain).hPslow(1)], ...
    {'Fast' 'Slow'});
pause(2)
%% Bootstrap
disp('Sampling dataset')
for tau = 1:nTauVals
    tauTxt = sprintf('tau %d', tau);
    disp(tauTxt)
    nDataPoints = length(strains(iStrain).overall(tau).disp);
    if useFractions == 1
        nSamples = round(fracData*nDataPoints);
    end
    strains(iStrain).overall(tau).samples = randi(nDataPoints, ...
        nSamples, nPopulations);
    strains(iStrain).samp(tau).name = sprintf(['tau' '%d'], tau);
    strains(iStrain).samp(tau).tval = tau*intTime;
    strains(iStrain).samp(tau).disp = strains(iStrain).overall(tau).disp(...
        strains(iStrain).overall(tau).samples);
```

```
strains(iStrain).samp(tau).uniq = nan(nSamples, nPopulations);
for iPop = 1:nPopulations
    strains(iStrain).samp(tau).uniq(1:length(unique(...
        strains(iStrain).samp(tau).disp(:,iPop))), iPop) = ...
        unique(strains(iStrain).samp(tau).disp(:,iPop));
    strains(iStrain).samp(tau).freq(:, iPop) = ...
        hist(strains(iStrain).samp(tau).disp(:, iPop), ...
        strains(iStrain).samp(tau).uniq(:, iPop));
end
strains(iStrain).samp(tau).prob = ...
    cumsum(strains(iStrain).samp(tau).freq)/nSamples;
strains(iStrain).samp(tau).mean = mean(strains(iStrain).samp(tau).prob);
strains(iStrain).samp(tau).stdv = std(strains(iStrain).samp(tau).prob);
if plotting == 1
    %% Plot sampled CPD curves
    strains(iStrain).samp(tau).figName = ...
        sprintf('%d ms (%s)', tau*frameTime, currStrainName);
    strains(iStrain).samp(tau).hF = figure('Color', 'w');
    strains(iStrain).samp(tau).hP = plot(...
        [strains(iStrain).samp(tau).uniq].^2, ...
        [strains(iStrain).samp(tau).prob]);
    titleText = sprintf('%s - Tau = %d ms; (%d groups of %d)', ...
        currStrainTitle, tau*frameTime, nPopulations, nSamples);
    title(titleText, 'FontSize', titleFontSize)
    strains(iStrain).samp(tau).hA = gca;
    set(strains(iStrain).samp(tau).hA, 'XLim', [1e-6 1], ...
        'YLim', [0 1], 'XScale', 'log', 'FontSize', labelFontSize)
    xlabel('squared step size (um^2)', 'FontSize', labelFontSize)
    yText = sprintf('P(r^2, %d ms)', tau*frameTime);
    ylabel(yText, 'FontSize', labelFontSize)
end
%% CPD fitting (bootstrap)
% Set up matrices to store results
strains(iStrain).samp(tau).resultParams((numParams + 1), nPopulations) = NaN;
strains(iStrain).samp(tau).residuals(nSamples, nPopulations) = NaN;
strains(iStrain).samp(tau).reducedChiSq(nSamples, nPopulations) = NaN;
strains(iStrain).samp(tau).resultingPoints(nSamples, nPopulations) = NaN;
strains(iStrain).samp(tau).prob(isnan(strains(iStrain).samp(tau).uniq)) = NaN;
disp('Calculating CPD fits')
% Sampled CPDs
for iPopn = 1:nPopulations
    FrameCounter(1, nPopulations, iPopn)
    goodFit = 0;
    badFitCounter = 1; %escape route for the while loop
    while goodFit == 0
        if sum(isnan(strains(iStrain).samp(tau).uniq(:, iPopn))) == 0
            uniqRows = nSamples;
        else
            uniqRows = find(isnan(...
                strains(iStrain).samp(tau).uniq(:, iPopn)), 1) - 1;
        end
        [resultParamListBoot, ~, ...
            strains(iStrain).samp(tau).residuals(1:uniqRows, iPopn)] = ...
            lsqcurvefit(@cpdFitFn, fitParams, ...
            strains(iStrain).samp(tau).uniq(1:uniqRows, iPopn).^2, ...
            strains(iStrain).samp(tau).prob(1:uniqRows, iPopn), ...
            fitLow, fitUp, options);
        % Transpose parameter list to fit in output matrix
        strains(iStrain).samp(tau).resultParams(:, iPopn) = ...
```

```
            resultParamListBoot';
        % Calculate reduced chi squared
        strains(iStrain).samp(tau).reducedChiSq(:, iPopn) = ...
            sum(strains(iStrain).samp(tau).residuals(:, iPopn).^2)/...
            (nSamples - numParams);
        % Sort parameters (alpha is fast, beta is slow)
        if strains(iStrain).samp(tau).resultParams(3, iPopn) < ...
            strains(iStrain).samp(tau).resultParams(4, iPopn)
            slowFrac = strains(iStrain).samp(tau).resultParams(1, iPopn);
            strains(iStrain).samp(tau).resultParams(1, iPopn) = ...
                strains(iStrain).samp(tau).resultParams(2, iPopn);
            strains(iStrain).samp(tau).resultParams(2, iPopn) = slowFrac;
            slowVal = strains(iStrain).samp(tau).resultParams(3, iPopn);
            strains(iStrain).samp(tau).resultParams(3, iPopn) = ...
                strains(iStrain).samp(tau).resultParams(4, iPopn);
            strains(iStrain).samp(tau).resultParams(4, iPopn) = slowVal;
        end
        % A 'good' fit has two distinct MSD values
        if strains(iStrain).samp(tau).resultParams(3, iPopn) <= ...
            2*strains(iStrain).samp(tau).resultParams(4, iPopn)
            goodFit = 0;
            badFitCounter = badFitCounter + 1;
        else
            goodFit = 1;
        end
        % Calculate fitted CPD
        strains(iStrain).samp(tau).resultingPoints(:, iPopn) = ...
            cpdFitFn(strains(iStrain).samp(tau).resultParams(:, iPopn), ...
            strains(iStrain).samp(tau).uniq(:, iPopn).^2);
        if badFitCounter > 9
            % wipe the parameters and move on
            strains(iStrain).samp(tau).resultParams(:, iPopn) = NaN;
            goodFit = 1;
        end
    end
end
numBadFits = sum(isnan(strains(iStrain).samp(tau).resultParams(1,:)));
numBadFitsMsg = sprintf('%d bad fits (of %d)', numBadFits, nPopulations);
disp(numBadFitsMsg)
%% Get stats on output parameters
strains(iStrain).samp(tau).paramMin = ...
    min(strains(iStrain).samp(tau).resultParams, [], 2);
strains(iStrain).samp(tau).paramMax = ...
    max(strains(iStrain).samp(tau).resultParams, [], 2);
strains(iStrain).samp(tau).paramMean = ...
    nanmean(strains(iStrain).samp(tau).resultParams, 2);
strains(iStrain).samp(tau).paramMedian = ...
    nanmedian(strains(iStrain).samp(tau).resultParams, 2);
strains(iStrain).samp(tau).paramStdv = ...
    nanstd(strains(iStrain).samp(tau).resultParams, [], 2);
if plotting == 1
    %% Plot bootstrapped fits
    strains(iStrain).samp(tau).nameFit = ...
        sprintf('%d ms (%s) fit', tau*frameTime, ...
        currStrainName);
    strains(iStrain).samp(tau).hFFit = figure('Color', 'w');
    strains(iStrain).samp(tau).hPFit = plot(...
        sort(strains(iStrain).samp(tau).uniq).^2, ...
        sort(strains(iStrain).samp(tau).resultingPoints));
```

```
        titleTextFit = sprintf('%s - Tau = %d ms; (%d groups of %d), fits', ...
            currStrainTitle, tau*frameTime, nPopulations, nSamples);
        title(titleTextFit, 'FontSize', titleFontSize)
        strains(iStrain).samp(tau).hAFit = gca;
        set(strains(iStrain).samp(tau).hAFit, 'XLim', [1e-6 1], ...
            'YLim', [0 1], 'XScale', 'log', 'FontSize', labelFontSize)
        xlabel('squared step size (um^2)', 'FontSize', labelFontSize)
        yText = sprintf('P(r^2, %d ms)', tau*frameTime);
        ylabel(yText, 'FontSize', labelFontSize)
    end
end
disp('Plotting MSDs from CPD fits')
%% Plot spread of MSDs
meanTitleText = sprintf('%s bootstrap MSDs', currStrainTitle);
strains(iStrain).msds.name = meanTitleText;
strains(iStrain).msds.hF = figure('Color', 'w');
%% Allocate space for the handle matrices
strains(iStrain).msds.hPfast(nPopulations, nTauVals) = 0;
strains(iStrain).msds.hPfastM(1, nTauVals) = 0;
strains(iStrain).msds.hPslow(nPopulations, nTauVals) = 0;
strains(iStrain).msds.hPslowM(1, nTauVals) = 0;
hold all
for tt = 1:nTauVals
    % Fast
    strains(iStrain).msds.hPfast(:, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).resultParams(3,:), 'r.');
    % Fast mean
    strains(iStrain).msds.hPfastM(1, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).paramMean(3,:), ...
        'gs', 'MarkerFaceColor', 'g');
    % Fast median
    strains(iStrain).msds.hPfastMed(1, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).paramMedian(3,:), ...
        'ks', 'MarkerFaceColor', 'm');
    % Slow
    strains(iStrain).msds.hPslow(:, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).resultParams(4,:), 'b.');
    % Slow mean
    strains(iStrain).msds.hPslowM(1, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).paramMean(4,:), ...
        'gs', 'MarkerFaceColor', 'g');
    % Slow median
    strains(iStrain).msds.hPslowMed(1, tt) = plot(...
        strains(iStrain).samp(tt).tval, ...
        strains(iStrain).samp(tt).paramMedian(4,:), ...
        'ks', 'MarkerFaceColor', 'm');
end
strains(iStrain).msds.hA = gca;
set(gca, 'XLim', [0 0.25])
xlabel('Tau (s)', 'FontSize', labelFontSize)
ylabel('MSD (um^2)', 'FontSize', labelFontSize)
title(meanTitleText, 'FontSize', titleFontSize)
legend([strains(iStrain).msds.hPfast(1) ...
        strains(iStrain).msds.hPslow(1) ...
```

```
            strains(iStrain).msds.hPfastM(1) ...
            strains(iStrain).msds.hPfastMed(1)], ...
            {'Fast' 'Slow' 'Means' 'Medians'})
    %% Plot mean MSDs with std dev error bars
    msdErrTitleTxt = sprintf('%s Mean MSDs (±1 std. dev.)', currStrainTitle);
    strains(iStrain).msdE.name = msdErrTitleTxt;
    strains(iStrain).msdE.hF = figure('Color', 'w');
    strains(iStrain).msdE.hP(nTauVals, 6) = 0;
    hold all
    for ttt = 1:nTauVals
        % Fast error bars (±1 std dev)
        strains(iStrain).msdE.hP(ttt, 5) = errorbar(...
            strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMean(3), ...
            strains(iStrain).samp(ttt).paramStdv(3), 'k');
        % Fast means
        strains(iStrain).msdE.hP(ttt, 1) = plot(...
            strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMean(3, :), ...
            'rs', 'MarkerFaceColor', 'r');
        % Fast medians
        strains(iStrain).msdE.hP(ttt, 3) = plot(...
        strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMedian(3, :), ...
            'rs', 'MarkerFaceColor', 'w');
        % Slow error bars (±1 std dev)
        strains(iStrain).msdE.hP(ttt, 6) = errorbar(...
            strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMean(4), ...
            strains(iStrain).samp(ttt).paramStdv(4), 'k');
        % Slow means
        strains(iStrain).msdE.hP(ttt, 2) = plot(...
            strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMean(4, :), ...
            'bs', 'MarkerFaceColor', 'b');
        % Slow medians
        strains(iStrain).msdE.hP(ttt, 4) = plot(...
            strains(iStrain).samp(ttt).tval, ...
            strains(iStrain).samp(ttt).paramMedian(4, :), ...
            'bs', 'MarkerFaceColor', 'w');
    end
    strains(iStrain).msdE.hA = gca;
    set(gca, 'XLim', [0 0.25])
    xlabel('Tau (s)', 'FontSize', labelFontSize)
    ylabel('MSD (um^2)', 'FontSize', labelFontSize)
    title(msdErrTitleTxt, 'FontSize', titleFontSize)
    legend(strains(iStrain).msdE.hP(1, 1:4), ...
        {'Fast (mean)' 'Slow (mean)' ...
         'Fast (median)' 'Slow (median)'})
end
%% Adjust y-axes
msdYLims(nStrains, 2) = 0;
meanMSDYLims(nStrains, 2) = 0;
for iiStrain = 1:nStrains
    msdYLims(iiStrain, :) = get(strains(iiStrain).msds.hA, 'YLim');
    meanMSDYLims(iiStrain, :) = get(strains(iiStrain).msdE.hA, 'YLim');
end
% MSDs
msdYLimit = [0 max(msdYLims(:, 2))];
```

```
% Mean MSDs
meanMSDYLimit = [0 max(meanMSDYLims(:, 2))];
for iType = 1:nStrains
    set(strains(iType).msds.hA, 'YLim', msdYLimit);
    set(strains(iType).msdE.hA, 'YLim', meanMSDYLimit);
end
%% Collect and plot fractions with standard deviations
fractions.all = figure('Color', 'w');
for cellType = 1:nStrains
    for tau = 1:nTauVals
        strains(cellType).alpha(tau) = strains(cellType).samp(tau).paramMean(1);
        strains(cellType).beta(tau) = strains(cellType).samp(tau).paramMean(2);
        strains(cellType).gamma(tau) = ...
            1 - sum(strains(cellType).samp(tau).paramMean(1:2));
        strains(cellType).MSDa(tau) = strains(cellType).samp(tau).paramMean(3);
        strains(cellType).MSDb(tau) = strains(cellType).samp(tau).paramMean(4);
        strains(cellType).stdAlpha(tau) = strains(cellType).samp(tau).paramStdv(1);
        strains(cellType).stdBeta(tau) = strains(cellType).samp(tau).paramStdv(2);
        strains(cellType).stdGamma(tau) = ...
            sqrt(strains(cellType).samp(tau).paramStdv(1)^2 + ...
            strains(cellType).samp(tau).paramStdv(2)^2);
        strains(cellType).stdMSDa(tau) = strains(cellType).samp(tau).paramStdv(3);
        strains(cellType).stdMSDb(tau) = strains(cellType).samp(tau).paramStdv(4);
        strains(cellType).meanAlpha = mean(strains(cellType).alpha);
        strains(cellType).meanBeta = mean(strains(cellType).beta);
        strains(cellType).meanGamma = mean(strains(cellType).gamma);
        strains(cellType).meanAlphaStd = ...
            sqrt(sum(strains(cellType).stdAlpha.^2))/nTauVals;
        strains(cellType).meanBetaStd = ...
            sqrt(sum(strains(cellType).stdBeta.^2))/nTauVals;
        strains(cellType).meanGammaStd = ...
            sqrt(sum(strains(cellType).stdGamma.^2))/nTauVals;
    end
    hold all
    errorbar(cellType, strains(cellType).meanAlpha, ...
        strains(cellType).meanAlphaStd, 'k');
    errorbar(cellType, strains(cellType).meanBeta, ...
        strains(cellType).meanBetaStd, 'k');
    errorbar(cellType, strains(cellType).meanGamma, ...
        strains(cellType).meanGammaStd, 'k');
    fractions.hA = plot(cellType, strains(cellType).meanAlpha, 'r.');
    fractions.hB = plot(cellType, strains(cellType).meanBeta, 'b.');
    fractions.hC = plot(cellType, strains(cellType).meanGamma, '.', ...
        'Color', [0 0.5 0]);
end
set(gca, 'XLim', [0 nStrains + 1])
set(gca, 'YLim', [0 1])
legend([fractions.hA fractions.hB fractions.hC], {'fast' 'slow' 'immobile'})
title('Fractions')
%% Mobile terms
fractions.mobile = figure('Color', 'w');
for cellKind = 1:nStrains
    hold all
    strains(cellKind).mobileFrac = ...
        sum([strains(cellKind).meanAlpha strains(cellKind).meanBeta]);
    strains(cellKind).mobileFracErr = sqrt(sum(...
        [strains(cellKind).meanAlphaStd strains(cellKind).meanBetaStd].^2));
    ARelErr = sqrt(sum([strains(cellKind).meanAlphaStd/strains(cellKind).meanAlpha ...
        strains(cellKind).mobileFracErr/strains(cellKind).mobileFrac].^2));
```

```
    BRelErr = sqrt(sum([strains(cellKind).meanBetaStd/strains(cellKind).meanBeta ...
        strains(cellKind).mobileFracErr/strains(cellKind).mobileFrac].^2));
    errorbar(cellKind, strains(cellKind).meanAlpha/...
        strains(cellKind).mobileFrac, ARelErr, 'k');
    errorbar(cellKind, strains(cellKind).meanBeta/...
        strains(cellKind).mobileFrac, BRelErr, 'k');
    fractions.hARel = plot(cellKind, strains(cellKind).meanAlpha/...
        strains(cellKind).mobileFrac, 'r.');
    fractions.hBRel = plot(cellKind, strains(cellKind).meanBeta/...
        strains(cellKind).mobileFrac, 'b.');
end
set(gca, 'XLim', [0 nStrains + 1])
set(gca, 'YLim', [0 1])
legend([fractions.hARel fractions.hBRel], {'fast' 'slow'})
title('Fraction of mobile term')
%% Immobile term
fractions.immobile = figure('Color', 'w');
for strainType = 1:nStrains
    hold all
    errorbar(cellType, strains(cellType).meanGamma, ...
        strains(cellType).meanGammaStd, 'k');
    fractions.hC = plot(cellType, strains(cellType).meanGamma, '.', ...
        'Color', [0 0.5 0]);
end
set(gca, 'XLim', [0 nStrains + 1])
set(gca, 'YLim', [0 0.5])
legend(fractions.hC(1), 'immobile')
title('Fraction of whole')
%% Save output
if useFractions == 1
    subFolderName = sprintf(['%d' 'pct'], fracData*100);
    fileName = sprintf(['bootstrap_cpd' '%d' '.mat'], fracData*100);
else
    subFolderName = sprintf(['%d' 'p ' '%d' 's'], nPopulations, nSamples);
    fileName = sprintf(['bootstrap_cpd' '%d' 's' '.mat'], nSamples);
end
outPath = [outFolder filesep subFolderName];
if ~exist(outPath, 'dir')
    mkdir(outPath)
end
fprintf('\n')
msgTxt3 = sprintf('Saving data to %s', fileName);
disp(msgTxt3)
% Data structure saved to 'mat' file
save([outPath filesep fileName], 'strains', 'fractions', '-v7.3');
% Figures
for iType = 1:nStrains
    figName = sprintf('%s whole dataset CPD', strains(iType).name);
    saveas(strains(iType).hF, [outPath filesep figName], 'fig');
    saveas(strains(iType).hF, [outPath filesep figName], 'jpg');
    figName2 = sprintf('%s sampled MSDs', strains(iType).name);
    saveas(strains(iType).msds.hF, [outPath filesep figName2], 'fig');
    saveas(strains(iType).msds.hF, [outPath filesep figName2], 'jpg');
    figName3 = sprintf('%s mean MSDs w std dev', strains(iType).name);
    saveas(strains(iType).msdE.hF, [outPath filesep figName3], 'fig');
    saveas(strains(iType).msdE.hF, [outPath filesep figName3], 'jpg');
    if plotting == 1
        for iiTau = 1:nTauVals
            figName4 = sprintf('%s sampled CPDs', strains(iType).name);
```

```
            saveas(strains(iType).samp(iiTau).hF, [outPath filesep figName4], 'fig');
            saveas(strains(iType).samp(iiTau).hF, [outPath filesep figName4], 'jpg');
            figName5 = sprintf('%s bootstrapped CPD fits', strains(iType).name);
            saveas(strains(iType).samp(iiTau).hFFit, ...
                [outPath filesep figName5], 'fig');
            saveas(strains(iType).samp(iiTau).hFFit, ...
                [outPath filesep figName5], 'jpg');
        end
    end
    figName6 = sprintf('Fractions');
    saveas(fractions.all, [outPath filesep figName6], 'fig');
    figName7 = sprintf('Mobile');
    saveas(fractions.mobile, [outPath filesep figName7], 'fig');
    figName8 = sprintf('Immobile');
    saveas(fractions.immobile, [outPath filesep figName8], 'fig');
end
outputFile = [outPath filesep fileName];
fprintf('\n')
disp('Done.')
%% CPD Fitting Function
    function cpdFit = cpdFitFn(parameters, rSquared)
        % cpdFit version 1.0.0
        % Based on |choose_function| by Yi Liao.
        numTerms = floor((length(parameters) - 1)/2) + 1; % 2 params/term
        sigmaSquared = (sigmaValue/1000)^2; % Convert nm to um^2
        fn = 0;
        usedFraction = 0;
        if numTerms == 1
            fn = exp(-rSquared/(parameters(1) + sigmaSquared));
            usedFraction = usedFraction + parameters(2); %should be 1
        else
            for currTerm = 1:numTerms - 1
                fn = fn + parameters(currTerm)*exp(-rSquared/...
                    (parameters(numTerms - 1 + currTerm) + sigmaSquared));
                usedFraction = usedFraction + parameters(currTerm);
            end
        end
        % Last term
        if mod(length(parameters) - 1, 2) == 0
            finalMsd = 0; %immobile
        else
            finalMsd = parameters(end - 1); %mobile
        end
        if length(parameters) - 1 > 1 %Don't add anything if it's just 1 mobile term
            parameters(end) = 1 - usedFraction;
            fn = fn + parameters(end)*exp(-rSquared/(finalMsd + sigmaSquared));
        end
        cpdFit = 1 - fn; %Fits to P(r^2, tau), i.e. 1 - C(r^2, tau)
    end
end
```

subplotFits

`subplotFits` is a diagnostic tool for checking the quality of the fits from `bootstrapCPD`. (Version 1.0.0)

- **Input:** the `*.mat` file generated by `bootstrapCPD`
- **Output:** figures of the fit parameters

```
function subplotFits(matFile)
load(matFile)
nStrains = size(strains, 2);
%% Options
A = [4 3 2 1]; %plotting time-lags 1-4, with first on top
titleTxt = {'enter strain names here'};
colorOrder = {'k' [0 0.7 0] 'r' 'b'};
% Output
[outPath, matName, ~] = fileparts(matFile);
[~, paramName, ~] = fileparts(outPath);
% paramName = 'M-I'
numMobile = str2double(paramName(1)); %M
numImmobile = str2double(paramName(3)); %I
numTerms = numMobile + numImmobile; %M+I
subFigRows = 5;
subFigCols = 5;
orderMatrix = {[1 6] [2 7] [11 16] [12 17]... %individual time-lags
               [3 8 13 18 23] ... %fast
               [4 9 14 19 24] ... %slow
               [5 10 15 20 25]... %immobile
               [21 22]}; %space for the legend
disp(matName)
s = struct;
for iStrain = 1:nStrains
    disp(strains(iStrain).name)
    s(iStrain).f = figure('Color', 'w', 'Position', [350 300 800 400]);
    hold all
    s(iStrain).h(5) = subplot(subFigRows, subFigCols, orderMatrix{5});
    s(iStrain).h1(4) = 0;
    if numTerms > 1
        s(iStrain).h2(4) = 0;
        s(iStrain).h(6) = subplot(subFigRows, subFigCols, orderMatrix{6});
    end
    if numTerms > 2
        s(iStrain).h3(4) = 0;
        s(iStrain).h(7) = subplot(subFigRows, subFigCols, orderMatrix{7});
    end
    if numImmobile == 1
        s(iStrain).h4(size(strains(1).samp(1).mean, 2), 4) = 0;
    end
    s(iStrain).gamma(4, size(strains(1).samp(1).mean, 2)) = 0;
    for iTau = 1:4
        %% Fast & slow at each tau
        s(iStrain).h(iTau) = subplot(subFigRows, subFigCols, orderMatrix{iTau});
        hold all
        s(iStrain).p1(iTau) = plot(strains(iStrain).samp(iTau).resultParams(1, :), ...
        strains(iStrain).samp(iTau).resultParams(1+numMobile, :), 'r.');
        s(iStrain).p2(iTau) = plot(strains(iStrain).samp(iTau).resultParams(2, :), ...
            strains(iStrain).samp(iTau).resultParams(2+numMobile, :), 'b.');
```

```
        if iTau == 3 || iTau == 4
            xlabel('fraction')
        end
        if iTau == 1 || iTau == 3
            ylabel('MSD')
        end
        if iTau == 3
            legend([s(iStrain).p1(3) s(iStrain).p2(3)], ...
                {'Fast' 'Slow'}, 'Location', [0.2725 0.09 0.0839 0.0813])
        end
        subTitleTxt = sprintf('%d ms', iTau*50);
        s(iStrain).sf(iTau) = title(subTitleTxt);
        %% Mobile/fast term panel
        axes(s(iStrain).h(5))
        hold all
        s(iStrain).h1(A(iTau)) = plot(...
            strains(iStrain).samp(A(iTau)).resultParams(1, :), ...
            strains(iStrain).samp(A(iTau)).resultParams(1+numMobile,:), ...
            '.', 'Color', colorOrder{iTau});
        %% Slow term panel
        axes(s(iStrain).h(6))
        hold all
        s(iStrain).h2(A(iTau)) = plot(...
            strains(iStrain).samp(A(iTau)).resultParams(2, :), ...
            strains(iStrain).samp(A(iTau)).resultParams(2+numMobile,:),...
            '.', 'Color', colorOrder{iTau});
        %% Immobile panel
        axes(s(iStrain).h(end))
        hold all
        s(iStrain).gamma(A(iTau), :) = 1 - ...
            sum(strains(iStrain).samp(A(iTau)).resultParams(1:numMobile,:));
        s(iStrain).h4(:, A(iTau)) = ...
            plot(s(iStrain).gamma(A(iTau), :), iTau, '.', ...
            'Color', colorOrder{iTau});
    end
    %% Label mobile/fast panel
    axes(s(iStrain).h(5));
    xlabel('fraction')
    ylabel('MSD')
    title('Fast')
    legend(s(iStrain).h1([1 2 3 4]), {'50 ms' '100 ms' '150 ms' '200 ms'})
    %% Label slow panel
    axes(s(iStrain).h(6)); %#ok<*LAXES>
    xlabel('fraction')
%   ylabel('MSD') %runs into the nearby panel
    title('Slow')
    %% Label immobile panel
    axes(s(iStrain).h(end));
    set(s(iStrain).h(end), 'XLim', [0 1], 'YLim', [0 5], ...
        'YTick', [0 5], 'YTickLabel', '')
    xlabel('fraction')
    title('Immobile')
    %% Make space for the legend
    s(iStrain).h(size(orderMatrix, 2)) = subplot(subFigRows, subFigCols, ...
        orderMatrix{end});
    axes(s(iStrain).h(end));
    axis off
    s(iStrain).fa = axes('Position',[0 0 1 1], 'XLim', [0 1], 'Ylim', [0 1],...
        'Box', 'off', 'Visible', 'off', 'Units', 'normalized', ...
```

```
        'clipping', 'off');
    text(0.5, 1, titleTxt{iStrain}, 'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'top')
end
%% Adjust axes
nFigs = size(orderMatrix, 2) - 1;
lastAxis = nFigs - 1;
yLimits(lastAxis) = 0;
xLimits(lastAxis) = 0;
for iFig = 1:lastAxis
    YLims(nStrains, (2*iFig - 1):2*iFig) = 0;
    XLims(nStrains, (2*iFig - 1):2*iFig) = 0;
    for iiStrain = 1:nStrains
        % y-axis
        YLims(iiStrain, (2*iFig - 1):2*iFig) = get(s(iiStrain).h(iFig), 'YLim');
        % x-axis
        XLims(iiStrain, (2*iFig - 1):2*iFig) = get(s(iiStrain).h(iFig), 'XLim');
    end
    yLimits(iFig) = max(YLims(:, 2*iFig));
    for strainNum = 1:nStrains
        set(s(iiStrain).h(iFig), 'YLim', [0 yLimits(iFig)]);
    end
    xLimits(iFig) = max(XLims(:, 2*iFig));
end
xLimits = [repmat(max(xLimits(1:4)), 1, 4) ...
           repmat(max(xLimits(5:lastAxis)), 1, lastAxis - 4)];
for iPlot = 1:lastAxis
    for iStrainNum = 1:nStrains
        set(s(iStrainNum).h(iPlot), 'XLim', [0 xLimits(iPlot)]);
    end
    if iPlot <= 4
        for iiStrainNum = 1:nStrains
            % title positions for the single tau subfigures
            set(s(iiStrainNum).sf(iPlot), 'Position', ...
                [0.5*xLimits(iPlot) 0.9*yLimits(iPlot) 0]);
        end
    end
end
s(4).yLim = [zeros(lastAxis, 1) yLimits'];
s(4).xLim = [zeros(lastAxis, 1) xLimits'];
for iType = 1:nStrains
    %% Save figures
    disp('Saving figures')
    figName = sprintf('%s MSDs and fractions', strains(iType).name);
    saveas(s(iType).f, [outPath filesep figName], 'fig');
    saveas(s(iType).f, [outPath filesep figName], 'jpg');
end
disp('Done.')
end
```

Simulations of diffusion on the cell membrane

simCell

`simCell` simulates a cell as a half cylinder lying along the y-axis. Molecules take a given number of steps in random directions. Steps are taken on the curved cylinder surface and then projected onto a two-dimensional plane. (Version 1.2.0)

- **Input:** cell type (*Vibrio cholerae* or *Bacteroides thetaiotaomicron*, for appropriate dimensions), step size in nm, integration time in ms, the number of steps to take, and the number of molecules to simulate
- **Output:** a data file of trajectories on the cylinder surface and the two-dimensional projection, as well as figures of the trajectories

```
function simCell(cellType, stepSize, intTime, nSteps, nMolecules)
versionNum = '1.2.0';
%% Parameters
if cellType == 1 %V. cholerae
    cellWidth = 0.6; %um (diameter)
    cellLength = 2; %um
else %B. theta
    cellWidth = 1; %um
    cellLength = 2.5; %um
end
pxSize = 49; %nm/px
%% Conversions
cellWidthPx = cellWidth*1000/pxSize; %px
cellLengthPx = cellLength*1000/pxSize; %px
stepSizePx = stepSize/pxSize; %px
intTimeS = intTime/1000; %s
cellRadius = cellWidthPx/2; %px; half the stdDev
curvedWidthPx = cellRadius*pi; %px; half the cell circumference
wBound = -curvedWidthPx/2; %"west" edge of cell
eBound = curvedWidthPx/2; %"east" edge of cell
nBound = cellLengthPx/2; %"north" edge of cell
sBound = -cellLengthPx/2; %"south" edge of cell
outFolder = uigetdir;
%% Prepare matrix for storing all track data
allData(nMolecules*nSteps, 9 + nErrCols) = 0;
lastRow = 0;
%% Choose trajectories
for iMolecule = 1:nMolecules
    xStart = unifrnd(wBound, eBound);
    yStart = unifrnd(sBound, nBound);
    xPrev = xStart;
    yPrev = yStart;
    %% Prepare matrix for storing data for the current molecule
    stepLog(nSteps, 9) = 0;
    xStartProj = cellRadius*sin(xStart/cellRadius);
    stepLog(1, 1:9) = [iMolecule 1 xStart yStart nan(1, 4) xStartProj];
    for iStep = 2:nSteps
        foundStep = 0;
        %% Choose steps
        while foundStep == 0
```

```
            xStep = stepSizePx*randn;
            yStep = stepSizePx*randn;
            % Convert to R and theta (for record keeping)
            R = sqrt(xStep^2+yStep^2);
            theta = tan(yStep/xStep);
            % Calculate new position in X & Y
            xNew = xPrev + xStep;
            yNew = yPrev + yStep;
            %% Keep matches
            if xNew <= eBound && xNew >= wBound && ...
                    yNew <= nBound && yNew >= sBound
                % Calculate x-position in projection
                xProj = cellRadius*sin(xNew/cellRadius);
                stepLog(iStep, 1:9) = [iMolecule iStep xNew yNew ...
                    R theta xStep yStep xProj];
                % Prepare to choose next step
                xPrev = xNew;
                yPrev = yNew;
                foundStep = 1;
            end
        end
    end
    %% Collect data together
    allData(lastRow + 1:lastRow + nSteps, :) = stepLog;
    lastRow = lastRow + nSteps;
end
%% Plot data
% Curved surface
f1 = figure('Color', 'w');
hold all
for iMol = 1:nMolecules
    plot(allData(allData(:, 1) == iMol, 3), ...
        allData(allData(:, 1) == iMol, 4), '-')
end
set(gca, 'XLim', [wBound eBound], 'YLim', [sBound nBound]);
axis equal
title('Paths on curved surface')
% 2D projection
f2 = figure('Color', 'w');
hold all
for iMol2 = 1:nMolecules
    plot(allData(allData(:, 1) == iMol2, 9), ...
        allData(allData(:, 1) == iMol2, 4), '-')
end
set(gca, 'XLim', [wBound eBound], 'YLim', [sBound nBound]);
axis equal
title('Projections of paths')
%% Save output
% Figures
saveas(f1, [outFolder filesep 'Curved paths'], 'fig');
saveas(f1, [outFolder filesep 'Curved paths'], 'jpg');
saveas(f2, [outFolder filesep 'Projections'], 'fig');
saveas(f2, [outFolder filesep 'Projections'], 'jpg');
% Data file
outFile = fopen([outFolder filesep 'Path data.dat'], 'wt');
% Headers
headerText = {'Track', 'Frame', 'X Curved', 'Y', 'R', 'Theta', ...
    'X Step', 'Y Step', 'X Proj'};
% Formats
```

```
headerFormat = ['%s' repmat(['\t' '%s'], 1, size(allData, 2) - 1), '\n'];
dataFormat = ['%g' repmat(['\t' '%g'], 1, size(allData, 2) - 1), '\n'];
% Save header
fprintf(outFile, headerFormat, headerText{:});
% Save data
fprintf(outFile, dataFormat, allData');
% Save parameters in footer
fprintf(outFile, 'Simulation created with simCell version %s\n', ...
    versionNum);
fprintf(outFile, 'Cell dimensions: %.1f um long, %.1f um wide.\n', ...
    cellLength, cellWidth);
fprintf(outFile, 'The simulated diffusion coefficient was %d', ...
    stepSize^2/(4*intTimeS));
fprintf(outFile, ['Boundaries are not sticky or reflective. ' ...
    'A new point is simply chosen that is within the boundaries.']);
fclose(outFile);
disp('Simulation complete.')
end
```

Bibliography

(2013). Artifacts of light. *Nature Methods*, 10(12):1135.

Abbe, E. (1873). Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung. *Archiv für Mikroskopische Anatomie*, 9(1):413–468.

Airy, G. B. (1835). On the diffraction of an object-glass with circular aperture. *Transactions of the Cambridge Philosophical Society*, 5:283–291.

Akrap, N., Seidel, T., and Barisas, B. G. (2010). Förster distances for fluorescent resonant energy transfer between mCherry and other visible fluorescent proteins. *Analytical Biochemistry*, 402(1):105–106.

Anderson, C. M., Georgiou, G. N., Morrison, I. E., Stevenson, G. V., and Cherry, R. J. (1992). Tracking of cell surface receptors by fluorescence digital imaging microscopy using a charge-coupled device camera. low-density lipoprotein and influenza virus receptor mobility at 4 degrees C. *Journal of Cell Science*, 101:415–425.

Anger, P., Bharadwaj, P., and Novotny, L. (2006). Enhancement and quenching of single-molecule fluorescence. *Physical Review Letters*, 96(11):113002.

Appelhans, T., Richter, C. P., Wilkens, V., Hess, S. T., Piehler, J., and Busch, K. B. (2012). Nanoscale organization of mitochondrial microcompartments revealed by combining tracking and localization microscopy. *Nano Letters*, 12(2):610–616.

Arvizo, R. R., Bhattacharyya, S., Kudgus, R. A., Giri, K., Bhattacharya, R., and Mukherjee, P. (2012). Intrinsic therapeutic applications of noble metal nanoparticles: past, present and future. *Chem Soc Rev*, 41(7):2943–70.

Baird, G. S., Zacharias, D. A., and Tsien, R. Y. (2000). Biochemistry, mutagenesis, and oligomerization of DsRed, a red fluorescent protein from coral. *Proceedings of the National Academy of Sciences USA*, 97(22):11984–11989.

Baker, H. and Bloom, W. L. (1948). Further studies on the Gram stain. *Journal of Bacteriology*, 56(4):387–390.

Bakshi, S., Siryaporn, A., Goulian, M., and Weisshaar, J. C. (2012). Superresolution imaging of ribosomes and RNA polymerase in live *Escherichia coli* cells. *Molecular Microbiology*, 85(1):21–38.

Bates, M., Huang, B., Dempsey, G. T., and Zhuang, X. (2007). Multicolor super-resolution imaging with photo-switchable fluorescent probes. *Science*, 317(5845):1749–1753.

Beck, N. A., Krukonis, E. S., and DiRita, V. J. (2004). TcpH influences virulence gene expression in *Vibrio cholerae* by inhibiting degradation of the transcription activator TcpP. *Journal of Bacteriology*, 186(24):8309–8316.

Bennish, M. L. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 15: Cholera: Pathophysiology, clinical features, and treatment, pages 229–255. American Society for Microbiology Press, Washington, DC.

Benson, R. C., Meyer, R. A., Zaruba, M. E., and McKhann, G. M. (1979). Cellular autofluorescence – Is it due to flavins? *Journal of Histochemistry & Cytochemistry*, 27(1):44–48.

Berg, O. G., Winter, R. B., and von Hippel, P. H. (1981). Diffusion-driven mechanisms of protein translocation on nucleic acids 1. Models and theory. *Biochemistry*, 20:6929–6948.

Berk, V., Fong, J. C., Dempsey, G. T., Develioglu, O. N., Zhuang, X., Liphardt, J., Yildiz, F. H., and Chu, S. (2012). Molecular architecture and assembly principles of *Vibrio cholerae* biofilms. *Science*, 337(6091):236–239.

Betzig, E. (1995). Proposed method for molecular optical imaging. *Optics Letters*, 20(3):237–239.

Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., Davidson, M. W., Lippincott-Schwartz, J., and Hess, H. F. (2006). Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793):1642–1645.

Biteen, J. S. (2012). Extending the tools of single-molecule fluorescence imaging to problems in microbiology. *Molecular Microbiology*, 85(1):1–4.

Biteen, J. S., Goley, E. D., Shapiro, L., and Moerner, W. E. (2012). Three-dimensional super-resolution imaging of the midplane protein FtsZ in live *Caulobacter crescentus* cells using astigmatism. *ChemPhysChem*, 13(4):1007–1012.

Biteen, J. S. and Moerner, W. E. (2010). Single-molecule and superresolution imaging in live bacteria cells. *Cold Spring Harbor Perspectives in Biology*, 2(3):a000448.

Biteen, J. S., Thompson, M. A., Tselentis, N. K., Bowman, G. R., Shapiro, L., and Moerner, W. E. (2008). Super-resolution imaging in live *Caulobacter crescentus* cells using photoswitchable EYFP. *Nature Methods*, 5(11):947–949.

Blanc-Potard, A.-B., Solomon, F., Kayser, J., and Groisman, E. A. (1999). The SPI-3 pathogenicity island of *Salmonella enterica*. *Journal of Bacteriology*, 181(3):998–1004.

Burnette, D. T., Sengupta, P., Dai, Y., Lippincott-Schwartz, J., and Kachar, B. (2011). Bleaching/blinking assisted localization microscopy for superresolution imaging using standard fluorescent molecules. *Proceedings of the National Academy of Sciences USA*, 108(52):21081–21086.

Carroll, P. A., Tashima, K. T., Rogers, M. B., DiRita, V. J., and Calderwood, S. B. (1997). Phase variation in *tcpH* modulates expression of the ToxR regulon in *Vibrio cholerae*. *Molecular Microbiology*, 25(6):1099–1111.

Centers for Disease Control and Prevention (2013). Cholera: General information. http://www.cdc.gov/cholera/general/.

Chalmers, N. I., Palmer, Jr, R. J., Du-Thumm, L., Sullivan, R., Shi, W., and Kolenbrander, P. E. (2007). Use of quantum dot luminescent probes to achieve single-cell resolution of human oral bacteria in biofilms. *Applied and Environmental Microbiology*, 73(2):630–636.

Charbon, G., Wang, J., Brustad, E., Schultz, P. G., Horwich, A. L., Jacobs-Wagner, C., and Chapman, E. (2011). Localization of GroEL determined by *in vivo* incorporation of a fluorescent amino acid. *Bioorganic & Medicinal Chemistry Letters*, 21(20):6067–6070.

Chen, I. and Ting, A. Y. (2005). Site-specific labeling of proteins with small molecules in live cells. *Current Opinion in Biotechnology*, 16:35–40.

Chin, J. W., Martin, A. B., King, D. S., Wang, L., and Schultz, P. G. (2002). Addition of a photocrosslinking amino acid to the genetic code of *Escherichia coli*. *Proceedings of the National Academy of Sciences USA*, 99(17):11020–11024.

Chiu, W., Baker, M. L., Jiang, W., Dougherty, M., and Schmid, M. F. (2005). Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure*, 13(3):363–72.

Chudakov, D. M., Feofanov, A. V., Mudrik, N. N., Lukyanov, S., and Lukyanov, K. A. (2003). Chromophore environment provides clue to "kindling fluorescent protein" riddle. *Journal of Biological Chemistry*, 278(9):7215–7219.

Chudakov, D. M., Matz, M., and Lukyanov, S. A. (2010). Fluorescent proteins and their applications in imaging living cells and tissues. *Physiological Reviews*, 90(3):1103–1163.

Churchman, L. S., Ökten, Z., Rock, R. S., Dawson, J. F., and Spudich, J. A. (2005). Single molecule high-resolution colocalization of Cy3 and Cy5 attached to macromolecules measures intramolecular distances through time. *Proceedings of the National Academy of Sciences USA*, 102(5):1419–1423.

Colwell, R. R. and Huq, A. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 9: Vibrios in the enivronment: Viable but nonculturable *Vibrio cholerae*, pages 117–133. American Society for Microbiology Press, Washington, DC.

Cravioto, A., Lanata, C. F., Lantagne, D. S., and Nair, G. B. (2011). Final report of the independent panel of experts on the cholera outbreak in Haiti. Technical report, United Nations Independent Panel.

Crawford, J. A., Krukonis, E. S., and DiRita, V. J. (2003). Membrane localization of the ToxR winged-helix domain is required for TcpP-mediated virulence gene activation in *Vibrio cholerae*. *Molecular Microbiology*, 47(5):1459–1473.

Dame, R. T. (2005). The role of nucleoid-associated proteins in the organization and compaction of bacterial chromatin. *Molecular Microbiology*, 56(4):858–870.

Dame, R. T., Kalmykowa, O. J., and Grainger, D. C. (2011). Chromosomal macrodomains and associated proteins: Implications for DNA organization and replication in Gram negative bacteria. *PLoS Genetics*, 7(6):e1002123.

De, S. N. (1959). Enterotoxicity of bacteria-free culture-filtrate of *Vibrio cholerae*. *Nature*, (4674):1533–1534.

Deich, J., Judd, E., McAdams, H., and Moerner, W. E. (2004). Visualization of the movement of single histidine kinase molecules in live *Caulobacter* cells. *Proceedings of the National Academy of Sciences USA*, 101(45):15921–15926.

Deiters, A., Cropp, T. A., Mukherji, M., Chin, J. W., Anderson, J. C., and Schultz, P. G. (2003). Adding amino acids with novel reactivity to the genetic code of *Saccharomyces cerevisiae*. *Journal of the American Chemical Society*, 125(39):11782–11783.

D'Elia, J. N. and Salyers, A. A. (1996). Effect of regulatory protein levels on utilization of starch by *Bacteroides thetaiotaomicron*. *Journal of Bacteriology*, 178(24):7180–7186.

Dell, C. L., Neely, M. N., and Olson, E. R. (1994). Altered pH lysine signalling mutants of *cadC*, a gene encoding a membrane-bound transcriptional activator of the *Escherichia coli cadBA* operon. *Molecular Microbiology*, 14(1):7–16.

Dertinger, T., Colyer, R. A., Iyer, G., Weiss, S., and Enderlein, J. (2009). Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI). *Proceedings of the National Academy of Sciences USA*, 106(52):22287–22292.

Di Rienzo, C., Gratton, E., Beltram, F., and Cardarelli, F. (2013). Fast spatiotemporal correlation spectroscopy to determine protein lateral diffusion laws in live cell membranes. *Proceedings of the National Academy of Sciences USA*, 110(30):12307–12312.

DiRita, V. J. and Mekalanos, J. J. (1991). Periplasmic interactions between two membrane regulatory proteins, ToxR and ToxS, results in signal transduction and transcriptional activation. *Cell*, 64(1):29–37.

Donehue, J. E., Haas, B. L., Wertz, E., Talicska, C. N., and Biteen, J. S. (2013). Plasmon-enhanced emission from single fluorescent proteins. *Proceedings of SPIE*, page 85970J.

Donehue, J. E., Wertz, E., Talicska, C. N., and Biteen, J. S. (2014). Plasmon-enhanced brightness and photostability from single fluorescent proteins coupled to gold nanorods. *submitted*.

Dupont, A. and Lamb, D. C. (2011). Nanoscale three-dimensional single particle tracking. *Nanoscale*, 3(11):4532–4541.

Durisic, N., Laparra-Cuervo, L., Sandoval-Álvarez, Á., Borbely, J. S., and Lakadamyali, M. (2014). Single-molecule evaluation of fluorescent protein photoactivation efficiency using an *in vivo* nanotemplate. *Nature Methods*, 11(2):156–162.

Egner, A., Geisler, C., von Middendorff, C., Bock, H., Wenzel, D., Medda, R., Andresen, M., Stiel, A. C., Jakobs, S., Eggeling, C., Schönle, A., and Hell, S. W. (2007). Fluorescence nanoscopy in whole cells by asynchronous localization of photoswitching emitters. *Biophysical Journal*, 93(9):3285–3290.

Einstein, A. (1905). Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322(8):549–560.

Elf, J., Li, G. W., and Xie, X. S. (2007). Probing transcription factor dynamics at the single-molecule level in a living cell. *Science*, 316:1191–1194.

Endesfelder, U. and Heilemann, M. (2014). Art and artifacts in single-molecule localization microscopy: beyond attractive images. *Nature Methods*, 11(3):235–238.

Endesfelder, U., Malkusch, S., Flottmann, B., Mondry, J., Liguzinski, P., Verveer, P. J., and Heilemann, M. (2011). Chemically induced photoswitching of fluorescent probes—A general concept for super-resolution microscopy. *Molecules*, 16(4):3106–3118.

Fernández-Suárez, M. and Ting, A. Y. (2008). Fluorescent probes for super-resolution imaging in living cells. *Nature Reviews Molecular Cell Biology*, 9(12):929–943.

Fischer, R. S., Wu, Y., Kanchanawong, P., Shroff, H., and Waterman, C. M. (2011). Microscopy in 3D: a biologist's toolbox. *Trends in Cell Biology*, 21(12):682–691.

Förster, T. (1948). Zwischenmolekulare Energiewanderung und Fluoreszenz. *Annalen der Physik*, 437(1):55–75.

Fu, Y., Zhang, J., and Lakowicz, J. R. (2008). Metal-enhanced fluorescence of single green fluorescent protein (GFP). *Biochemical and Biophysical Research Communications*, 376(4):712–717.

Ghosh, A., Paul, K., and Chowdhury, R. (2006). Role of the histone-like nucleoid structuring protein in colonization, motility, and bile-dependent repression of virulence gene expression in *Vibrio cholerae*. *Infection and Immunity*, 74(5):3060–3064.

Giannone, G., Hosy, E., Levet, F., Constals, A., Schulze, K., Sobolevsky, A. I., Rosconi, M. P., Gouaux, E., Tampé, R., Choquet, D., and Cognet, L. (2010). Dynamic superresolution imaging of endogenous proteins on living cells at ultra-high density. *Biophysical Journal*, 99(4):1303–1310.

Giepmans, B. N. G., Adams, S. R., Ellisman, M. H., and Tsien, R. Y. (2006). The fluorescent toolbox for assessing protein location and function. *Science*, 312(5771):217–224.

Givaty, O. and Levy, Y. (2009). Protein sliding along DNA: Dynamics and structural characterization. *Journal of Molecular Biology*, 385(4):1087–1097.

Gordon, M. P., Ha, T., and Selvin, P. R. (2004). Single-molecule high-resolution imaging with photobleaching. *Proceedings of the National Academy of Sciences USA*, 101(17):6462–6465.

Goss, T. J., Morgan, S. J., French, E. L., and Krukonis, E. S. (2013). Toxr recognizes a direct repeat element in the *toxT*, *ompU*, *ompT*, and *ctxA* promoters of *Vibrio cholerae* to regulate transcription. *Infection and Immunity*, 81(3):884–895.

Goss, T. J., Seaborn, C. P., Gray, M. D., and Krukonis, E. S. (2010). Identification of the TcpP-binding site in the *toxT* promoter of *Vibrio cholerae* and the role of ToxR in TcpP-mediated activation. *Infection and Immunity*, 78(10):4122–4133.

Graham, L. L., Harris, R., and Beveridge, T. J. (1991). Freeze-substitution of Gram-negative eubacteria: General cell morphology and envelope profiles. *Journal of Bacteriology*, 173(5):1623–1633.

Griffin, B. A., Adams, S. R., and Tsien, R. Y. (1998). Specific covalent labeling of recombinant protein molecules inside live cells. *Science*, 281(5374):269–272.

Gurskaya, N. G., Verkhusha, V. V., Shcheglov, A. S., Staroverov, D. B., Chepurnykh, T. V., Fradkov, A. F., Lukyanov, S. A., and Lukyanov, K. A. (2006). Engineering of a monomeric green-to-red photoactivatable fluorescent protein induced by blue light. *Nature Biotechnology*, 24(4):461–465.

Gustafsson, M. G. L. (2005). Nonlinear structured-illumination microscopy: Wide-field fluorescence imaging with theoretically unlimited resolution. *Proceedings of the National Academy of Sciences USA*, 102(37):13081–13086.

Guzman, L. M., Belin, D., Carson, M. J., and Beckwith, J. (1995). Tight regulation, modulation, and high-level expression by vectors containing the arabinose PBAD promoter. *Journal of Bacteriology*, 177(14):4121–4130.

Habuchi, S., Ando, R., Dedecker, P., Verheijen, W., Mizuno, H., Miyawaki, A., and Hofkens, J. (2005). Reversible single-molecule photoswitching in the GFP-like fluorescent protein Dronpa. *Proceedings of the National Academy of Sciences USA*, 102(27):9511–9516.

Hall, R. H. (2011). A De in the life of cholera. *Indian Journal of Medical Research*, 133:146–152.

Hammar, P., Leroy, P., Mahmutovic, A., Marklund, E. G., Berg, O. G., and Elf, J. (2012). The *lac* repressor displays facilitated diffusion in living cells. *Science*, 336(6088):1595–1598.

Hao, Q., Qiu, T., and Chu, P. K. (2012). Surface-enhanced cellular fluorescence imaging. *Progress in Surface Science*, 87(1-4):23–45.

Häse, C. C. and Mekalanos, J. J. (1998). TcpP protein is a positive regulator of virulence gene expression in *Vibrio cholerae*. *Proceedings of the National Academy of Sciences USA*, 95(2):730–734.

He, R.-Y., Chang, G.-L., Wu, H.-L., Lin, C.-H., Chiu, K.-C., Su, Y.-D., and Chen, S.-J. (2006). Enhanced live cell membrane imaging using surface plasmon-enhanced total internal reflection fluorescence microscopy. *Optics Express*, 14(20):9307–9316.

Hebert, B., Costantino, S., and Wiseman, P. W. (2005). Spatiotemporal image correlation spectroscopy (STICS) theory, verification, and application to protein velocity mapping in living CHO cells. *Biophysical Journal*, 88(5):3601–3614.

Heilemann, M., van de Linde, S., Schüttpelz, M., Kasper, R., Seefeldt, B., Mukherjee, A., Tinnefeld, P., and Sauer, M. (2008). Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes. *Angewandte Chemie International Edition*, 47(33):6172–6176.

Hell, S. W. (2007). Far-field optical nanoscopy. *Science*, 316:1153–1158.

Hell, S. W. and Wichmann, J. (1994). Breaking the diffraction resolution limit by stimulated emission: Stimulated-emission-depletion fluorescence microscopy. *Optics Letters*, 19(11):780–782.

Henriques, R., Griffiths, C., Hesper Rego, E., and Mhlanga, M. M. (2011). PALM and STORM: Unlocking live-cell super-resolution. *Biopolymers*, 95(5):322–331.

Hess, S. T., Girirajan, T. P. K., and Mason, M. D. (2006). Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical Journal*, 91(11):4258–4272.

Higgins, D. E. and DiRita, V. J. (1994). Transcriptional control of *toxT*, a regulatory gene in the ToxR regulon of *Vibrio cholerae*. *Molecular Microbiology*, 14(1):17–29.

Hu, C.-D. and Kerppola, T. K. (2003). Simultaneous visualization of multiple protein interactions in living cells using multicolor fluorescence complementation analysis. *Nature Biotechnology*, 21(5):539–545.

Huang, B., Bates, M., and Zhuang, X. (2009). Super-resolution fluorescence microscopy. *Annual Reviews of Biochemistry*, 78(1):993–1016.

Huang, B., Jones, S. A., Brandenburg, B., and Zhuang, X. (2008). Whole-cell 3D STORM reveals interactions between cellular structures with nanometer-scale resolution. *Nature Methods*, 5(12):1047–1052.

Huang, F., Schwartz, S. L., Byars, J. M., and Lidke, K. A. (2011). Simultaneous multiple-emitter fitting for single molecule super-resolution imaging. *Biomedical Optics Express*, 2(5):1377–1393.

Jaqaman, K. and Danuser, G. (2009). Computational image analysis of cellular dynamics: A case study based on particle tracking. *Cold Spring Harbor Protocols*, 4(12):pdb.top65.

Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., and Danuser, G. (2008). Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, 5(8):695–702.

Jose, J., Jordan, L. R., Johnson, T. W., Lee, S. H., Wittenberg, N. J., and Oh, S.-H. (2013). Topographically flat substrates with embedded nanoplasmonic devices for biosensing. *Advanced Functional Materials*, 23(22):2812–2820.

Kaper, J. B., Fasano, A., and Trucksis, M. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 11: Toxins of *Vibrio cholerae*, pages 145–176. American Society for Microbiology Press, Washington, DC.

Karunatilaka, K. S., Cameron, E. A., Martens, E. C., Koropatkin, N. M., and Biteen, J. S. (2014). Super-resolution imaging captures carbohydrate utilization dynamics in human gut symbionts. *submitted*.

Kay, B. A., Bopp, C. A., and Wells, J. G. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 1: Isolation and Identification of *Vibrio cholerae* O1 from Fecal Specimens, pages 3–26. American Society for Microbiology Press, Washington, DC.

Kaysner, C. and Hill, W. E. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 2: Toxigenic *Vibrio cholerae* O1 in Food and Water, pages 27–39. American Society for Microbiology Press, Washington, DC.

Kim, S. Y., Gitai, Z., Kinkhabwala, A., Shapiro, L., and Moerner, W. E. (2006). Single molecules of the bacterial actin MreB undergo directed treadmilling motion in *Caulobacter crescentus*. *Proceedings of the National Academy of Sciences USA*, 103(29):10929–10934.

Koch, A. L. (1996). What size should a bacterium be? A question of scale. *Annual Review of Microbiology*, 50:317–348.

Kolibachuk, D. and Greenberg, E. P. (1993). The *Vibrio fischeri* luminescence gene activator LuxR is a membrane-associated protein. *Journal of Bacteriology*, 175(22):7307–7312.

Koropatkin, N. M., Martens, E. C., Gordon, J. I., and Smith, T. J. (2008). Starch catabolism by a prominent human gut symbiont is directed by the recognition of amylose helices. *Structure*, 16(7):1105–1115.

Krukonis, E. S., Yu, R. R., and DiRita, V. J. (2000). The *Vibrio cholerae* ToxR/TcpP/ToxT virulence cascade: Distinct roles for two membrane-localized transcriptional activators on a single promoter. *Molecular Microbiology*, 38(1):67–84.

Kubitscheck, U., Kückmann, O., Kues, T., and Peters, R. (2000). Imaging and tracking of single GFP molecules in solution. *Biophysical Journal*, 78(4):2170–2179.

Kumar, M., Mommer, M. S., and Sourjik, V. (2010). Mobility of cytoplasmic, membrane, and DNA-binding proteins in *Escherichia coli*. *Biophysical Journal*, 98(4):552–559.

Lassak, K., Peeters, E., Wróbel, S., and Albers, S.-V. (2013). The one-component system ArnR: A membrane-bound activator of the crenarchaeal archaellum. *Molecular Microbiology*, 88(1):125–139.

Le Moal, E., Fort, E., Lévêque-Fort, S., Cordelières, F. P., Fontaine-Aupart, M. P., and Ricolleau, C. (2007). Enhanced fluorescence cell imaging with metal-coated slides. *Biophysical Journal*, 92(6):2150–2161.

Lee, H.-l. D., Sahl, S. J., Lew, M. D., and Moerner, W. E. (2012). The double-helix microscope super-resolves extended biological structures by localizing single blinking molecules in three dimensions with nanoscale precision. *Applied Physics Letters*, 100:153701.

Lee, S. F., Thompson, M. A., Schwartz, M. A., Shapiro, L., and Moerner, W. E. (2011). Super-resolution imaging of the nucleoid-associated protein HU in *Caulobacter crescentus*. *Biophysical Journal*, 100(7):L31–L33.

Levene, M. J., Korlach, J., Turnder, S. W., Foquet, M., Craighead, H. G., and Webb, W. W. (2003). Zero-mode waveguides for single-molecule analysis at high concentrations. *Science*, 299:682–686.

Lew, M. D., Lee, S. F., Ptacin, J. L., Lee, M. K., Twieg, R. J., Shapiro, L., and Moerner, W. E. (2011). Three-dimensional superresolution colocalization of intracellular protein superstructures and the cell surface in live *Caulobacter crescentus*. *Proceedings of the National Academy of Sciences USA*, 108(46):E1102–E1110.

Li, C. C., Crawford, J. A., and DiRita, V. J. (2000). Molecular cloning and transcriptional regulation of *ompT*, a toxr-repressed gene in *Vibrio cholerae*. *Molecular Microbiology*, 35(1):189–203.

Lin, Z., Kumagai, K., Baba, K., Mekalanos, J. J., and Nishibuchi, M. (1993). *Vibrio parahaemolyticus* has a homolog of the *Vibrio cholerae toxRS* operon that mediates environmentally induced regulation of the thermostable direct hemolysin gene. *Journal of Bacteriology*, 175(12):3844–3855.

Lohmüller, T., Triffo, S., O'Donoghue, G. P., Xu, Q., Coyle, M. P., and Groves, J. T. (2011). Supported membranes embedded with fixed arrays of gold nanoparticles. *Nano Letters*, 11:4912–4918.

Lommerse, P. H., Blab, G. A., Cognet, L., Harms, G. S., Snaar-Jagalska, E., Spaink, H. P., and Schmidt, T. (2004). Single-molecule imaging of the H-Ras membrane-anchor

reveals domains in the cytoplasmic leaflet of the cell membrane. *Biophysical Journal*, 86(1):609–616.

Los, G. V., Encell, L. P., McDougall, M. G., Hartzell, D. D., Karassina, N., Zimprich, C., Wood, M. G., Learish, R., Ohana, R. F., Urh, M., Simpson, D., Mendez, J., Zimmerman, K., Otto, P., Vidugiris, G., Zhu, J., Darzins, A., Klaubert, D. H., Bulleit, R. F., and Wood, K. V. (2008). HaloTag: A novel protein labeling technology for cell imaging and protein analysis. *ACS Chemical Biology*, 3(6):373–382.

Magde, D., Elson, E., and Webb, W. W. (1972). Thermodynamic fluctuations in a reacting system—Measurement by fluorescence correlation spectroscopy. *Physical Review Letters*, 29:705–708.

Mahler, B., Spinicelli, P., Buil, S., Quelin, X., Hermier, J.-P., and Dubertret, B. (2008). Towards non-blinking colloidal quantum dots. *Nature Materials*, 7(8):659–664.

Matson, J. S., Withey, J. H., and DiRita, V. J. (2007). Regulatory networks controlling *Vibrio cholerae* virulence gene expression. *Infection and Immunity*, 75(12):5542–5549.

Matz, M. V., Fradkov, A. F., Labas, Y. A., Savitsky, A. P., Zaraisky, A. G., Markelov, M. L., and Lukyanov, S. A. (1999). Fluorescent proteins from nonbioluminescent Anthozoa species. *Nature Biotechnology*, 17(10):969–973.

Mekalanos, J. J. (2011). *Epidemiological and Molecular Aspects on Cholera*, chapter 6. The evolution of *Vibrio cholerae* as a pathogen, pages 97–114. Springer, New York.

Merrell, D. S. and Camilli, A. (2000). Regulation of *Vibrio cholerae* genes required for acid tolerance by a member of the "ToxR-like" family of transcriptional regulators. *Journal of Bacteriology*, 182(19):5342–5350.

Michalet, X. (2011). Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. *Physical Review E*, 82(4):041914.

Michalet, X. and Berglund, A. J. (2012). Optimal diffusion coefficient estimation in single-particle tracking. *Physical Review E*, 85(6):061916.

Michalet, X., Pinaud, F. F., Bentolila, L. A., Tsay, J. M., Doose, S., Li, J. J., Sundaresan, G., Wu, A. M., Gambhir, S. S., and Weiss, S. (2005). Quantum dots for live cells, *in vivo* imaging, and diagnostics. *Science*, 307(5709):538–544.

Miller, V. L., Taylor, R. K., and Mekalanos, J. J. (1987). Cholera toxin transcriptional activator ToxR is a transmembrane DNA binding protein. *Cell*, 48(2):271–279.

Moerner, W. and Orrit, M. (1999). Illuminating single molecules in condensed matter. *Science*, 283:1670–1676.

Moerner, W. E. (2012). Microscopy beyond the diffraction limit using actively controlled single molecules. *Journal of Microscopy*, 246(3):213–220.

Moolman, M. C., Huang, Z., Krishnan, S. T., Kerssemakers, J. W. J., and Dekker, N. H. (2013). Electron beam fabrication of a microfluidic device for studying submicron-scale bacteria. *Journal of Nanobiotechnology*, 11:12.

Morgan, S. J., Felek, S., Gadwal, S., Koropatkin, N. M., Perry, J. W., Bryson, A. B., and Krukonis, E. S. (2011). The two faces of ToxR: activator of *ompU*, co-regulator of *toxT* in *Vibrio cholerae*. *Molecular Microbiology*, 81(1):113–128.

Morris, Jr., J. G. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 7: *Vibrio cholerae* O139 Bengal, pages 95–102. American Society for Microbiology Press, Washington, DC.

Murley, Y. M., Carroll, P. A., Skorupski, K., Taylor, R. K., and Calderwood, S. B. (1999). Differential transcription of the *tcpPH* operon confers biotype-specific control of the *Vibrio cholerae* ToxR virulence regulon. *Infection and Immunity*, 67(10):5117–5123.

Mutavdžić, D., Xu, J., Thakur, G., Triulzi, R., Kasas, S., Jeremić, M., Leblanc, R., and Radotić, K. (2011). Determination of the size of quantum dots by fluorescence spectroscopy. *Analyst*, 136(11):2391–2396.

Nagai, T., Ibata, K., Park, E. S., Kubota, M., Mikoshiba, K., and Miyawaki, A. (2002). A variant of yellow fluorescent protein with fast and efficient maturation for cell-biological applications. *Nature Biotechnology*, 20(1):87–90.

Narayanan, J., Xiong, J.-Y., and Liu, X.-Y. (2006). Determination of agarose gel pore size: Absorbance measurements *vis a vis* other techniques. In *Journal of Physics: Conference Series*, volume 28 of *International Conference on Materails for Advanced Technologies, Symposium Y*, pages 83–86. Institute of Physics Publishing.

Neely, M. N., Dell, C. L., and Olson, E. R. (1994). Roles of LysP and CadC in mediating the lysine requirement for acid induction of the *Escherichia coli cad* operon. *Journal of Bacteriology*, 176(11):3278–3285.

Nye, M. B., Pfau, J. D., Skorupski, K., and Taylor, R. K. (2000). *Vibrio cholerae* H-NS silences virulence gene expression at multiple steps in the ToxR regulatory cascade. *Journal of Bacteriology*, 182(15):4295–4303.

Nye, M. B. and Taylor, R. K. (2003). *Vibrio cholerae* H-NS domain structure and function with respect to transcriptional repression of ToxR regulon genes reveals differences among H-NS family members. *Molecular Microbiology*, 50(2):427–444.

Orfanides, P., Buckner, T. F., and Buncick, M. C. (2000). Demonstration of surface plasmons in metal island films and the effect of the surrounding medium—An undergraduate experiment. *American Journal of Physics*, 68(10):986–942.

Ormö, M., Cubitt, A. B., Kallio, K., Gross, L. A., Tsien, R. Y., and Remington, S. J. (1996). Crystal structure of the *Aequorea victoria* green fluorescent protein. *Science*, 273(5820):1392–1395.

Pavani, S. R. P., Thompson, M. A., Biteen, J. S., Lord, S. J., Liu, N., Twieg, R. J., Piestun, R., and Moerner, W. E. (2009). Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function. *Proceedings of the National Academy of Sciences USA*, 106(9):2995–2999.

Peters, I. M., de Grooth, B. G., Schins, J. M., Figdor, C. G., and Greve, J. (1998). Three dimensional single-particle tracking with nanometer resolution. *Review of Scientific Instruments*, 69(7):2762–2766.

Peters, R., Peters, J., Tews, K. H., and Bähr, W. (1974). A microfluorimetric study of translational diffusion in erythrocyte membranes. *Biochimica et Biophysica Acta*, 367(3):282–294.

Pinaud, F. and Dahan, M. (2011). Targeting and imaging single biomolecules in living cells by complementation-activated light microscopy with split-fluorescent proteins. *Proceedings of the National Academy of Sciences USA*, 108(24):E201–E210.

Qian, H., Sheetz, M. P., and Elson, E. L. (1991). Single particle tracking. *Biophysical Journal*, 60:910–921.

Qiu, Y., Chen, X., Li, Y., Zheng, B., Li, S., Chen, W. R., and Liu, H. (2012). Impact of the optical depth of field on cytogenetic image quality. *Journal of Biomedical Optics*, 17(9):096017.

Qu, X., Wu, D., Mets, L., and Scherer, N. F. (2004). Nanometer-localized multiple single-molecule fluorescence microscopy. *Proceedings of the National Academy of Sciences USA*, 101(31):11298–11303.

Reich, K. A. and Schoolnik, G. K. (1994). The light organ symbiont *Vibrio fischeri* possesses a homolog of the *Vibrio cholerae* transmembrane transcriptional activator ToxR. *Journal of Bacteriology*, 176(10):3085–3088.

Ritchie, K., Lill, Y., Sood, C., Lee, H., and Zhang, S. (2013). Single-molecule imaging in live bacteria cells. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 368(1611):20120355.

Robson, A., Burrage, K., and Leake, M. C. (2012). Inferring diffusion in single live cells at the single molecule level. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 368(1611):20120029.

Rodrigue, D. C., Popovic, T., and Wachsmuth, I. K. (1994). *Vibrio cholerae and cholera: Molecular to global perspectives*, chapter 5: Nontoxigenic *Vibrio cholerae* O1 Infections in the United States, pages 69–76. American Society for Microbiology Press, Washington, DC.

Rosselló-Mora, R. and Amann, R. (2001). The species concept for prokaryotes. *FEMS Microbiology Reviews*, 25:39–67.

Rowland, D. J. and Biteen, J. S. (2014). Top-hat and asymmetric Gaussian-based fitting functions for quantifying directional single-molecule motion. *ChemPhysChem*, 15(4):712–720.

Ruemmele, J. A., Hall, W. P., Ruvuna, L. K., and Van Duyne, R. P. (2013). A localized surface plasmon resonance imaging instrument for multiplexed biosensing. *Analytical Chemistry*, 85:4560–4566.

Rust, M. J., Bates, M., and Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature Methods*, 3(10):793–795.

Savage, D. C. (1977). Microbial ecology of the gastrointestinal tract. *Annual Review of Microbiology*, 31:107–133.

Saxton, M. J. (1997). Single-particle tracking: The distribution of diffusion coefficients. *Biophysical Journal*, 72(4):1744–1753.

Saxton, M. J. (2007). *Methods in Molecular Biology*, volume 400: Methods in Membrane Lipids, chapter 20: Modeling 2D and 3D Diffusion, pages 295–321. Humana Press, Totowa, NJ.

Schoen, I., Ries, J., Klotzsch, E., Ewers, H., and Vogel, V. (2011). Binding-activated localization microscopy of DNA structures. *Nano Letters*, 11:4008–4011.

Schütz, G. J., Schindler, H., and Schmidt, T. (1997). Single-molecule microscopy on model membranes reveals anomalous diffusion. *Biophysical Journal*, 73:1073–1080.

Semrau, S. and Schmidt, T. (2007). Particle image correlation spectroscopy (PICS): Retrieving nanometer-scale correlations from high-density single-molecule position data. *Biophysical Journal*, 92(2):613–621.

Shaner, N. C., Campbell, R. E., Steinbach, P. A., Gipemans, B. N. G., Palmer, A. E., and Tsien, R. Y. (2004). Improved monomeric red, orange and yellow fluorescent proteins derived from *Discosoma* sp. red fluorescent protein. *Nature Biotechnology*, 22(12):1567–1572.

Shaner, N. C., Lin, M. Z., McKeown, M. R., Steinbach, P. A., Hazelwood, K. L., Davidson, M. W., and Tsien, R. Y. (2008). Improving the photostability of bright monomeric orange and red fluorescent proteins. *Nature Methods*, 5(6):545–551.

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21.

Sharonov, A. and Hochstrasser, R. M. (2006). Wide-field subdiffraction imaging by accumulated binding of diffusing probes. *Proceedings of the National Academy of Sciences USA*, 103(50):18911–18916.

Shroff, H., Galbraith, C. G., Galbraith, J. A., and Betzig, E. (2008). Live-cell photoactivated localization microscopy of nanoscale adhesion dynamics. *Nature Methods*, 5(5):417–423.

Shuang, B., Byers, C. P., Kisley, L., Wang, L.-Y., Zhao, J., Morimura, H., Link, S., and Landes, C. F. (2013). Improved analysis for determining diffusion coefficients from short, single-molecule trajectories with photoblinking. *Langmuir*, 29(1):228–234.

Sochacki, K. A., Shkel, I. A., Record, M. T., and Weisshaar, J. C. (2011). Protein diffusion in the periplasm of *E. coli* under osmotic stress. *Biophysical Journal*, 100(1):22–31.

Sonnleitner, A., Schütz, G. J., and Schmidt, T. (1999). Free Brownian motion of individual lipid molecules in biomembranes. *Biophysical Journal*, 77(5):2638–2642.

Stonehouse, E. A., Hulbert, R. R., Nye, M. B., Skorupski, K., and Taylor, R. K. (2011). H-NS binding and repression of the *ctx* promoter in *Vibrio cholerae*. *Journal of Bacteriology*, 193(4):979–988.

Subach, F. V., Malashkevich, V. N., Zencheck, W. D., Xiao, H., Filonov, G. S., Almo, S. C., and Verkhusha, V. V. (2009a). Photoactivation mechanism of PAmCherry based on crystal structures of the protein in the dark and fluorescent states. *Proceedings of the National Academy of Sciences USA*, 106(50):21097–21102.

Subach, F. V., Patterson, G. H., Manley, S., Gillette, J. M., Lippincott-Schwartz, J., and Verkhusha, V. V. (2009b). Photoactivatable mCherry for high-resolution two-color fluorescence microscopy. *Nature Methods*, 6(2):153–159.

Sundaramurthy, A., Schuck, P. J., Kino, G. S., and Moerner, W. E. (2004). Gap-dependent optical coupling of single "bowtie" nanoantennas resonant in the visible. *Nano Letters*, 4(5):957–961.

Taminiau, T. H., Stefani, F. D., Segerink, F. B., and van Hulst, N. F. (2008). Optical antennas direct single-molecule emission. *Nature Photonics*, 2(4):234–237.

Tan, B. J. Y., Sow, C. H., Koh, T. S., Chin, K. C., Wee, A. T. S., and Ong, C. K. (2005). Fabrication of size-tunable gold nanoparticles array with nanosphere lithography, reactive ion etching and thermal annealing. *Journal of Physical Chemistry B*, 109:11100–11109.

Tanaka, K. A. K., Suzuki, K. G. N., Shirai, Y. M., Shibutani, S. T., Miyahara, M. S. H., Tsuboi, H., Yahara, M., Yoshimura, A., Mayor, S., Fujiwara, T. K., and Kusumi, A. (2010). Membrane molecules mobile even after chemical fixation. *Nature Methods*, 7(11):865–866.

Thompson, R. E., Larson, D. R., and Webb, W. W. (2002). Precise nanometer localization analysis for individual fluorescent probes. *Biophysical Journal*, 82(5):2775–2783.

Tsien, R. Y. (1998). The green fluorescent protein. *Annual Reviews of Biochemistry*, 67:509–544.

van den Wildenberg, S. M. J. L., Bollen, Y. J. M., and Peterman, E. J. G. (2011). How to quantify protein diffusion in the bacterial membrane. *Biopolymers*, 95(5):312–321.

van Oijen, A. M., Köhler, J., and Schmidt, J. (1999). Far-field fluorescence microscopy beyond the diffraction limit. *Journal of the Optical Society of America A*, 16(4):909–915.

Verkhusha, V. V. and Lukyanov, K. A. (2004). The molecular properties and applications of Anthozoa fluorescent proteins and chromoproteins. *Nature Biotechnology*, 22(3):289–296.

Wagner, M., Weber, P., Bruns, T., Strauss, W. S., Wittig, R., and Schneckenburger, H. (2010). Light dose is a limiting factor to maintain cell viability in fluorescence microscopy and single molecule detection. *International Journal of Molecular Sciences*, 11(3):956–966.

Wang, W., Li, G.-W., Chen, C., Xie, X. S., and Zhuang, X. (2011). Chromosome organization by a nucleoid-associated protein in live bacteria. *Science*, 333(6048):1445–1449.

Warmack, R. J. and Humphrey, S. L. (1986). Observation of two surface-plasmon modes on gold particles. *Physical Review B*, 34:2246–2252.

Welch, T. J. and Bartlett, D. H. (1998). Identification of a regulatory protein required for pressure-responsive gene expression in the deep-sea bacterium *Photobacterium* species strain SS9. *Molecular Microbiology*, 27(5):977–985.

Welsher, K. and Yang, H. (2014). Multi-resolution 3D visualization of the early stages of cellular uptake of peptide-coated nanoparticles. *Nature Nanotechnology*, 9(3):198–203.

Willets, K. A., Nishimura, S. Y., Schuck, P. J., Twieg, R. J., and Moerner, W. (2005). Nonlinear optical chromophores as nanoscale emitters for single-molecule spectroscopy. *Accounts of Chemical Research*, 38(7):549–556.

Willets, K. A. and Van Duyne, R. P. (2007). Localized surface plasmon resonance spectroscopy and sensing. *Annual Review of Physical Chemistry*, 58(1):267–297.

Withey, J. H. and DiRita, V. J. (2006). The toxbox: Specific DNA sequence requirements for activation of *Vibrio cholerae* virulence genes by ToxT. *Molecular Microbiology*, 59(6):1779–1789.

Xia, T., Li, N., and Fang, X. (2013). Single-molecule fluorescence imaging in living cells. *Annual Review of Physical Chemistry*, 64(1):459–480.

Xia, Y. and Halas, N. J. (2005). Shape-controlled synthesis and surface plasmonic properties of metallic nanostructures. *MRS Bulletin*, 30:338–348.

Xiao, J., Elf, J., Li, G.-W., Yu, J., and Xie, X. (2008). *Single-Molecule Techniques: A Laboratory Manual*, chapter 7: Imaging Gene Expression in Living Cells at the Single-Molecule Level, pages 149–169. Cold Spring Harbor Laboratory Press.

Xie, X. S., Choi, P. J., Li, G.-W., Lee, N. K., and Lia, G. (2008). Single-molecule approach to molecular biology in living bacterial cells. *Annual Review of Biophysics*, 37(1):417–444.

Xu, K., Babcock, H. P., and Zhuang, X. (2012). Dual-objective STORM reveals three-dimensional filament organization in the actin cytoskeleton. *Nature Methods*, 9(2):1–6.

Yang, Y. and Isberg, R. R. (1997). Transcriptional regulation of the *Yersinia pseudotuberculosis* pH 6 antigen adhesin by two envelope-associated components. *Molecular Microbiology*, 24(3):499–510.

Yildiz, A., Forkey, J. N., McKinney, S. A., Ha, T., Goldman, Y. E., and Selvin, P. R. (2003). Myosin V walks hand-over-hand: Single fluorophore imaging with 1.5-nm localization. *Science*, 300(5628):2061–2065.

Yu, J., Xiao, J., Ren, X., Lao, K., and Xie, X. S. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767):1600–1603.

Yu, R. R. and DiRita, V. J. (1999). Analysis of an autoregulatory loop controlling ToxT, cholera toxin, and toxin-coregulated pilus production in *Vibrio cholerae*. *Journal of Bacteriology*, 181(8):2584–2592.

Yu, R. R. and DiRita, V. J. (2002). Regulation of gene expression in *Vibrio cholerae* by ToxT involves both antirepression and RNA polymerase stimulation. *Molecular Microbiology*, 43(1):119–134.

Zareh, S. K., DeSantis, M. C., Kessler, J. M., Li, J.-L., and Wang, Y. M. (2012). Single-image diffusion coefficient measurements of proteins in free solution. *Biophysical Journal*, 102(7):1685–1691.

Zhang, R.-G., Scott, D. L., Westbrook, M. L., Nance, S., Spangler, B. D., Shipley, G. G., and Westbrook, E. M. (1995). The three-dimensional crystal structure of cholera toxin. *Journal of Molecular Biology*, 251:563–573.

Zhang, X., Hicks, E. M., Zhao, J., Schatz, G. C., and Van Duyne, R. P. (2005). Electrochemical tuning of silver nanoparticles fabricated by nanosphere lithography. *Nano Letters*, 5(7):1503–1507.

Zhang, Z., Smith, B. A. C., Wang, L., Brock, A., Cho, C., and Schultz, P. G. (2003). A new strategy for the site-specific modification of proteins *in vivo*. *Biochemistry*, 42(22):6735–6746.