Determining epitope-specificity of TCRs with Gaussian processes

Aalto University
School of Science
Department of Computer Science

Emmi Jokinen
22.3.2019

Outline

Immunity, T cells and T cell receptors
Motivation and objectives
Sequence data
Kernel methods
Gaussian processes

Outline

Immunity, T cells and T cell receptors

Motivation and objectives

Sequence data

Kernel methods

Gaussian processes

Human immune system

- Humans are exposed to millions of potential pathogens daily, through contact, ingestion, and inhalation.

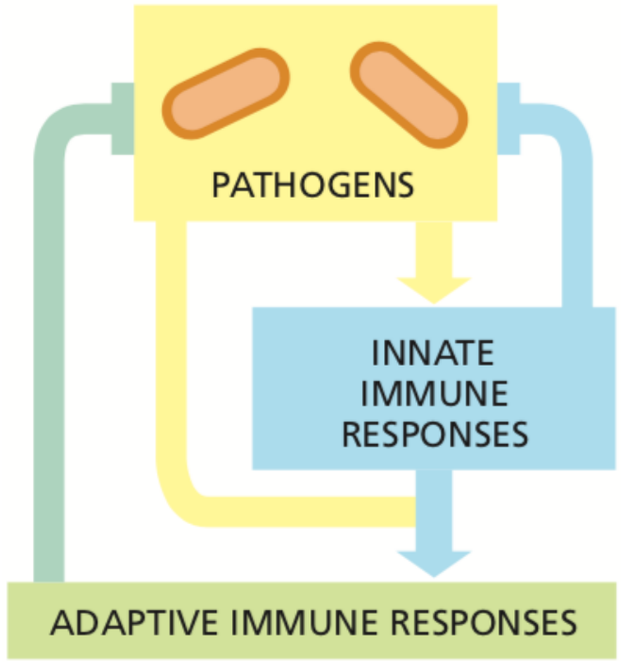


Figure: [1]

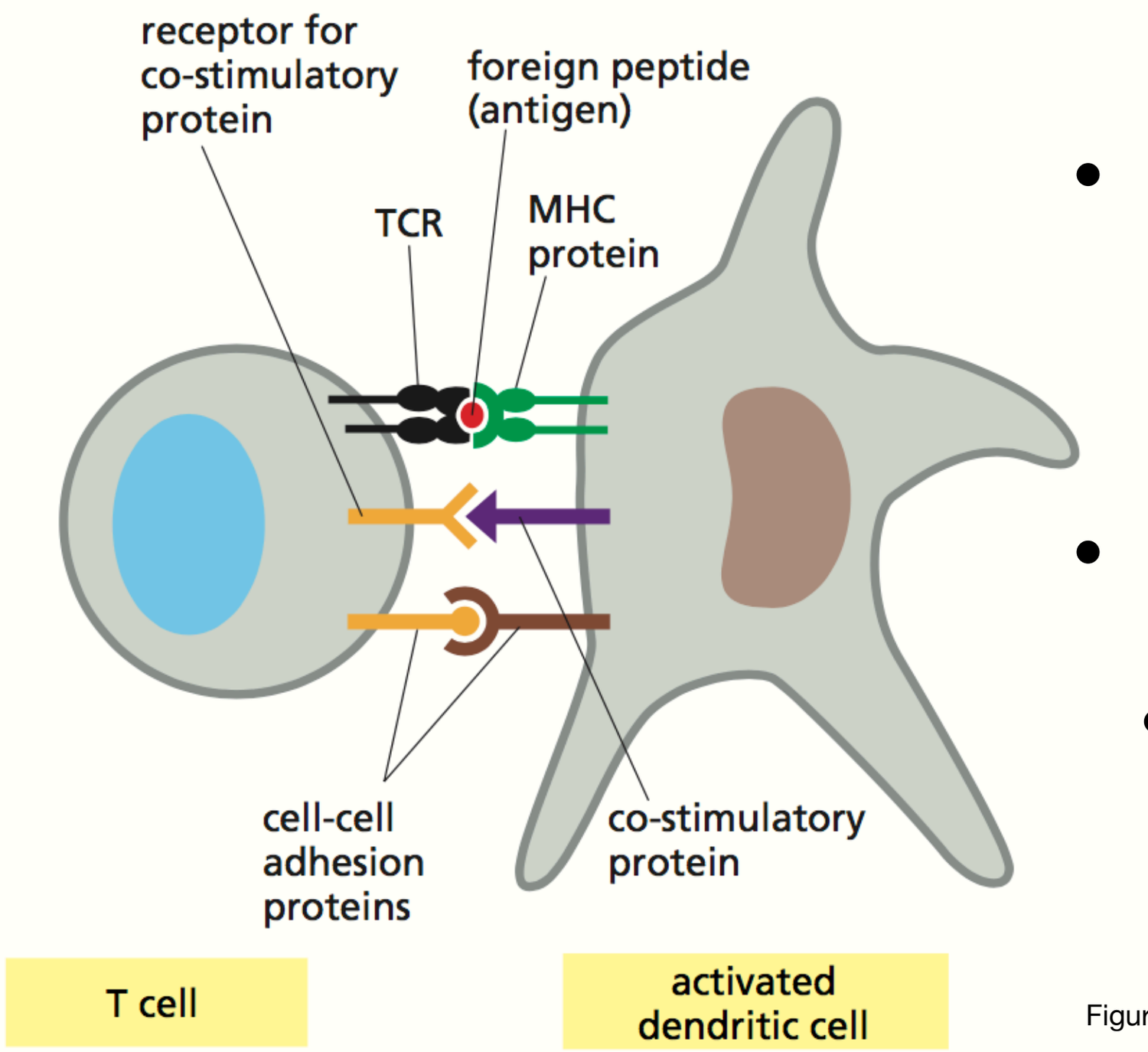
Innate Immune responses

- General defence reactions
- Three lines of defences:
  - physical and chemical barriers
  - cell-intrinsic responses
    - an individual cell recognizes that it has been infected and takes measures to kill or cripple the invader
  - a specialized set of proteins and phagocytic cells that recognize conserved features of pathogens and become quickly activated to help destroy invaders

Adaptive immune responses

- Highly specific responses
- slow to develop on first exposure to a new pathogen (can take a week or so)
- Provide long-term protection
- Activated by innate immune system
- Carried out by lymphocytes
  - Antibody responses (B cells)
  - **T-cell-mediated responses**

T cells and T cell receptors (TCRs)



- Peptides are displayed by MHC proteins located on the surface of antigen-presenting cells (usually dendritic cells)
- T cells are activated by foreign antigens
  - proliferate and differentiate into effector cells

Figure: [1]

Antigen–binding site

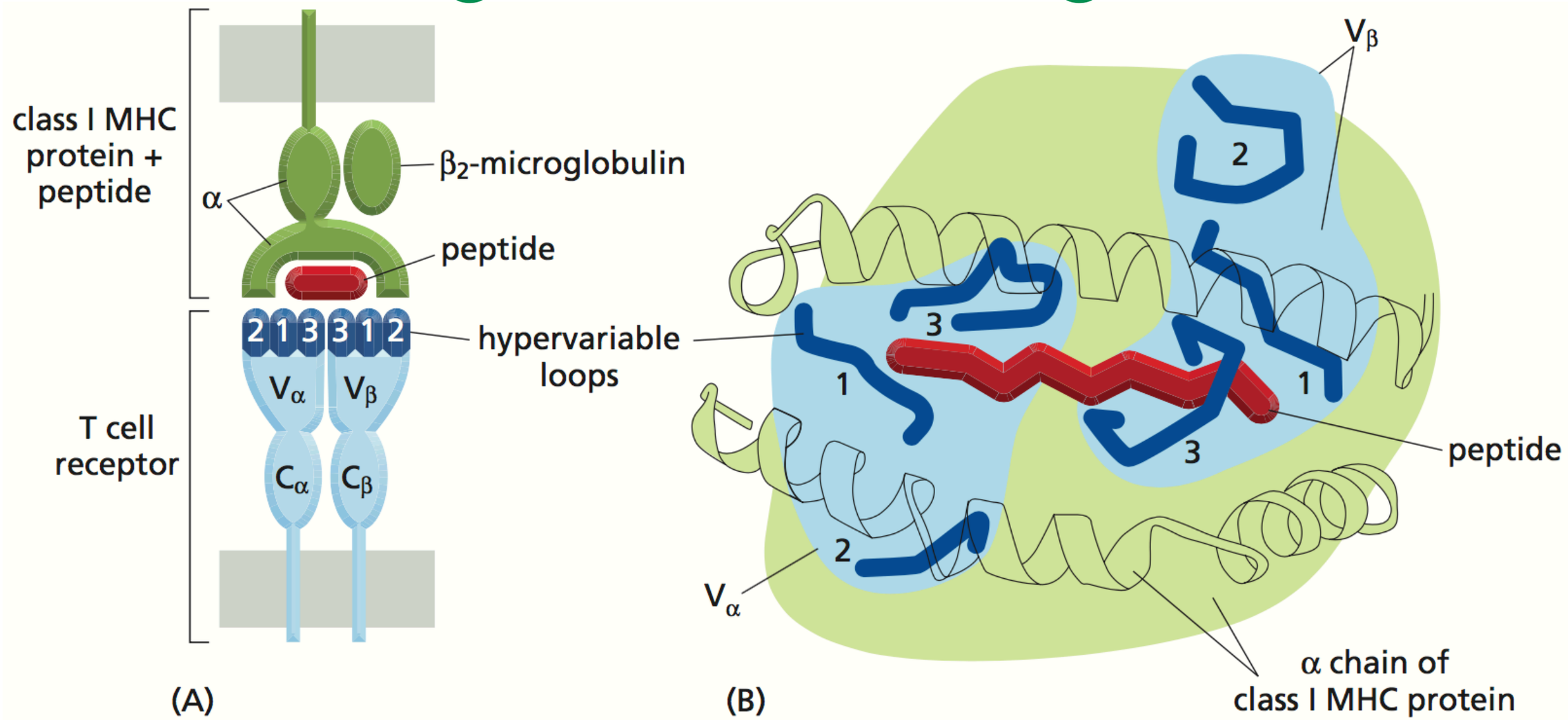
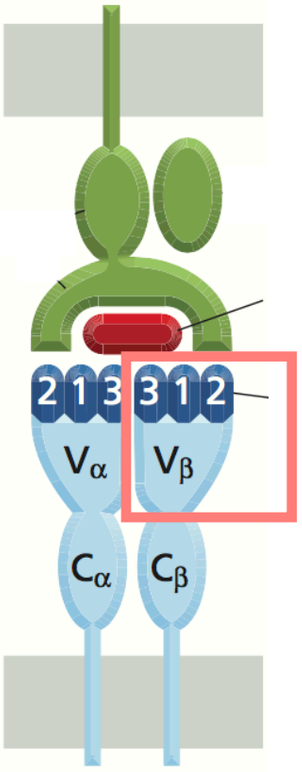
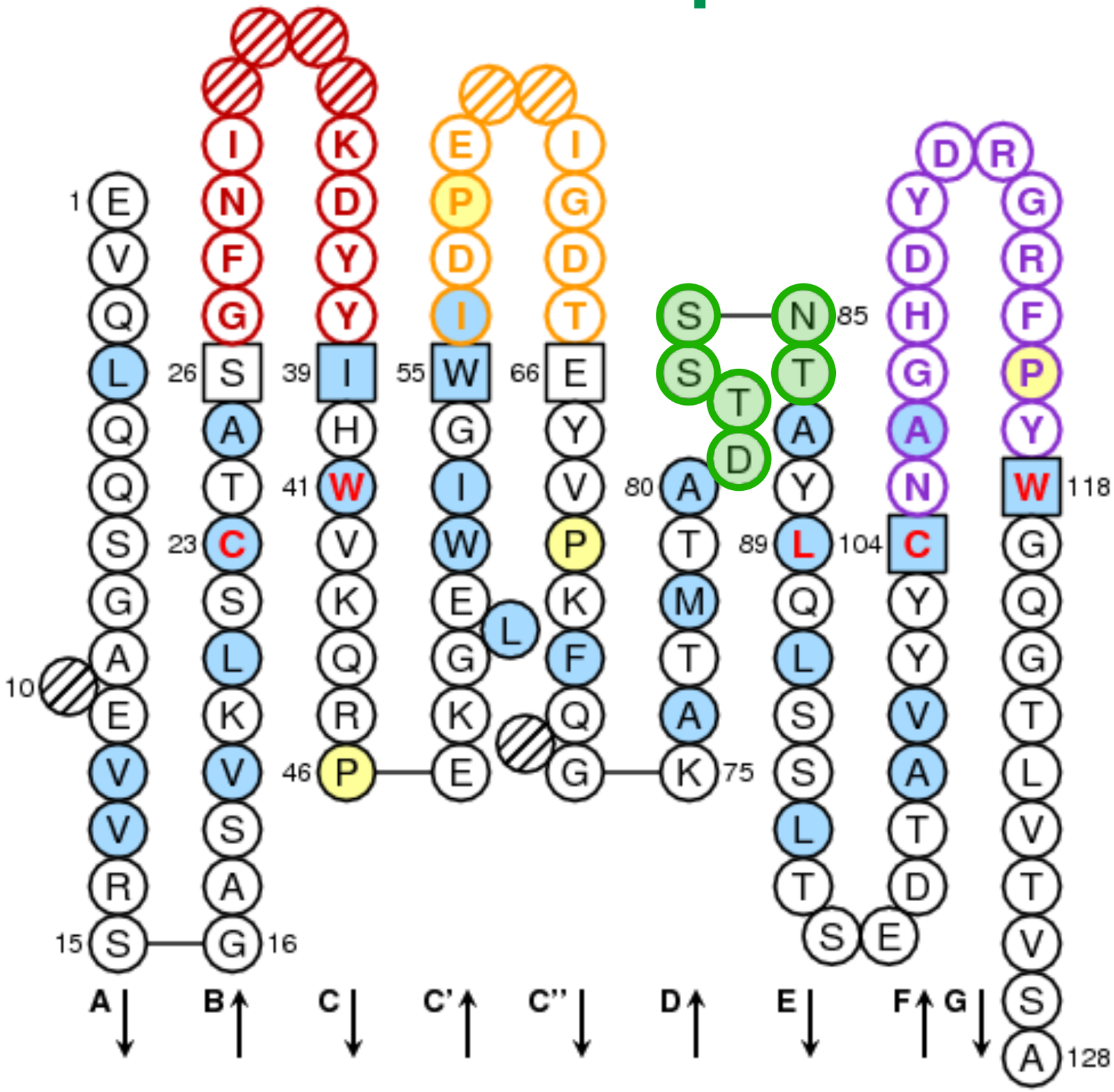


Figure: [1]

- CDR3 primarily interacts with the peptide and is most variable
- CDR1 and CDR2 (and CDR2.5) mainly bind to the walls of the peptide-binding groove, but have sometimes been observed to be in contact with the peptide



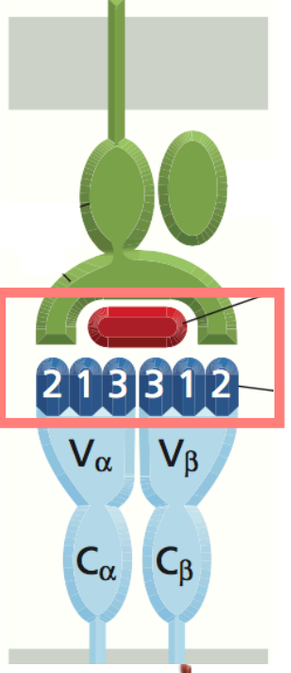
Collier-de-Perles presentation of TCRβ



CDR2.5 has sometimes been observed to make contact with the peptide

Figure: http://www.imgt.org/3Dstructure-DB/doc/IMGTCollier-de-Perles.shtml (modified)

TCR + pMHC



CDR3α
CDR2.5α
CDR2α
CDR1α
CDR3β
CDR2.5β
CDR2β
CDR1β

peptide: NLVPMVATV (pp65)

Peptide binding groove of MHC

Observed contacts from 52 crystal structures

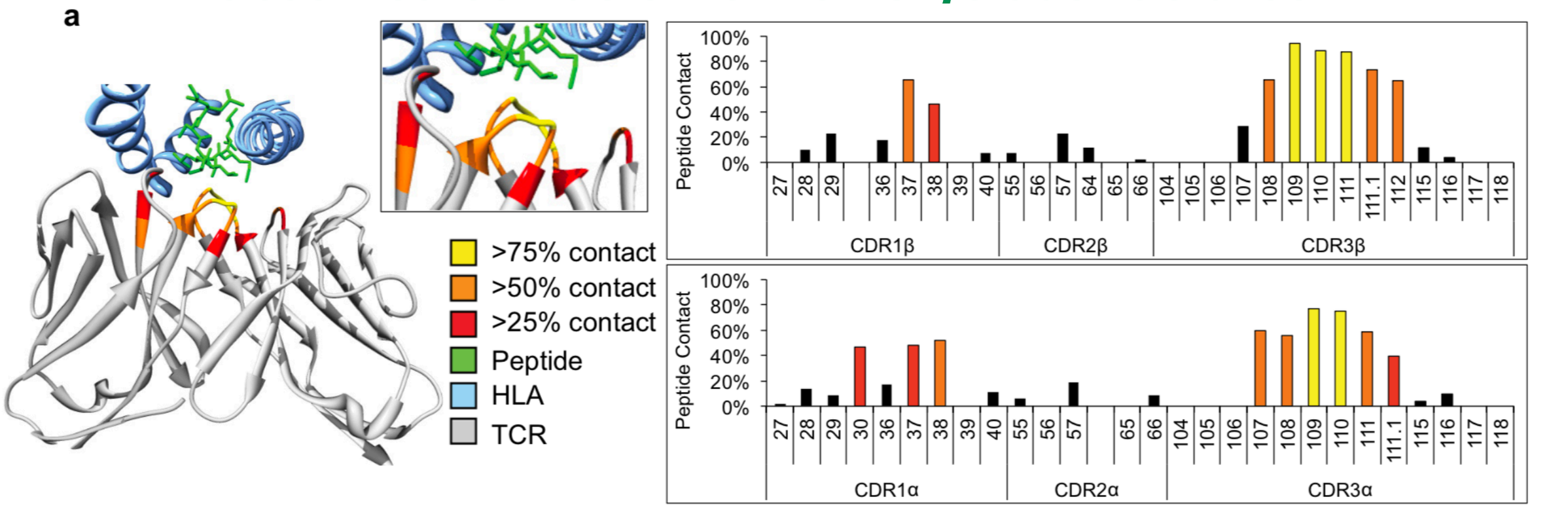


Figure: Identifying specificity groups in the T cell receptor repertoire. Glanville, J., et al., 2017

TCR diversity

- **V(D)J recombination:** TCRs are manufactured from variable (V), diversity (D), joining (J) and constant (C) gene fragments through a process of somatic gene rearrangement
- Humans have
  - ~50 V-gene, 2 D-gene and 12 J-gene segments in TCRβ chain locus
  - 45 V-gene and 50 J-gene segments in TCRα chain locus
- **Junctional diversification:** During the joining of these gene segments nucleotides can be lost from the ends of the segments, and one or more can also be inserted
  - It has been estimated that V(D)J recombination can result to around 10^{18} different TCRs

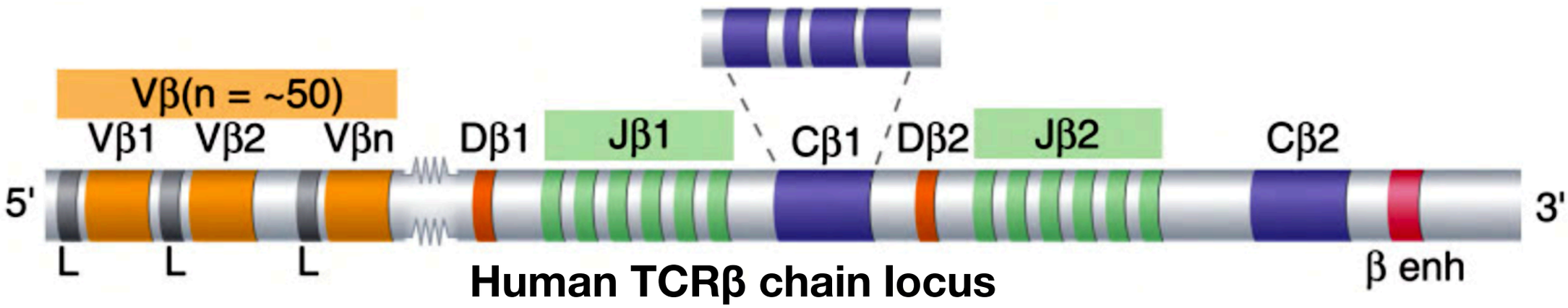
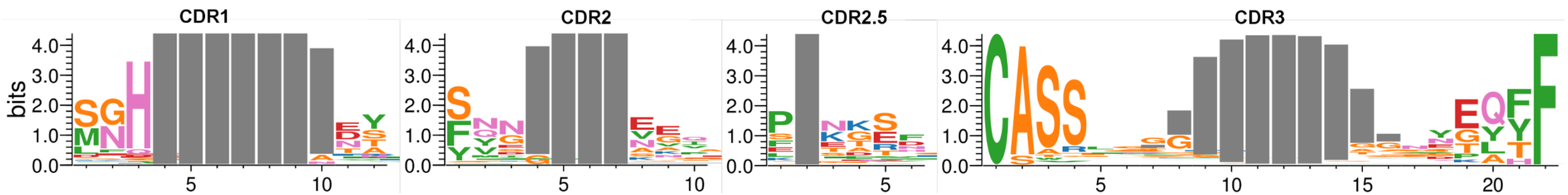


Figure: Cellular and Molecular Immunology. Abul K. Abbas, Andrew H. H. Lichtman, Shiv

Complexity of TCR repertoires

- ~10^{18} possible TCRs
  - ~10^{12} T cells in a human
    - ~10^{8} distinct TCRs in a person (young adult)
    - If a sample contains e.g. around 50 000 T cells
      - It's about 0.000005 % of all T cells

- On average, each T cell recognises at least 1 million individual peptides
- A peptide can be recognised by several TCRs.



Sequence logo of 412 TCRβ sequences that recognize peptide NLVPMVATV

TCR repertoire

- TCRs of an individual are called a TCR repertoire
- Each T cell has potentially unique TCR
- After a T cell has recognized an epitope, it starts to proliferate
  - The resulting set of T cells with identical TCRs is called a clone
    - T cells from large clones are more likely to be sampled

Outline

Immunity, T cells and T cell receptors

Motivation and objectives

Sequence data

Kernel methods

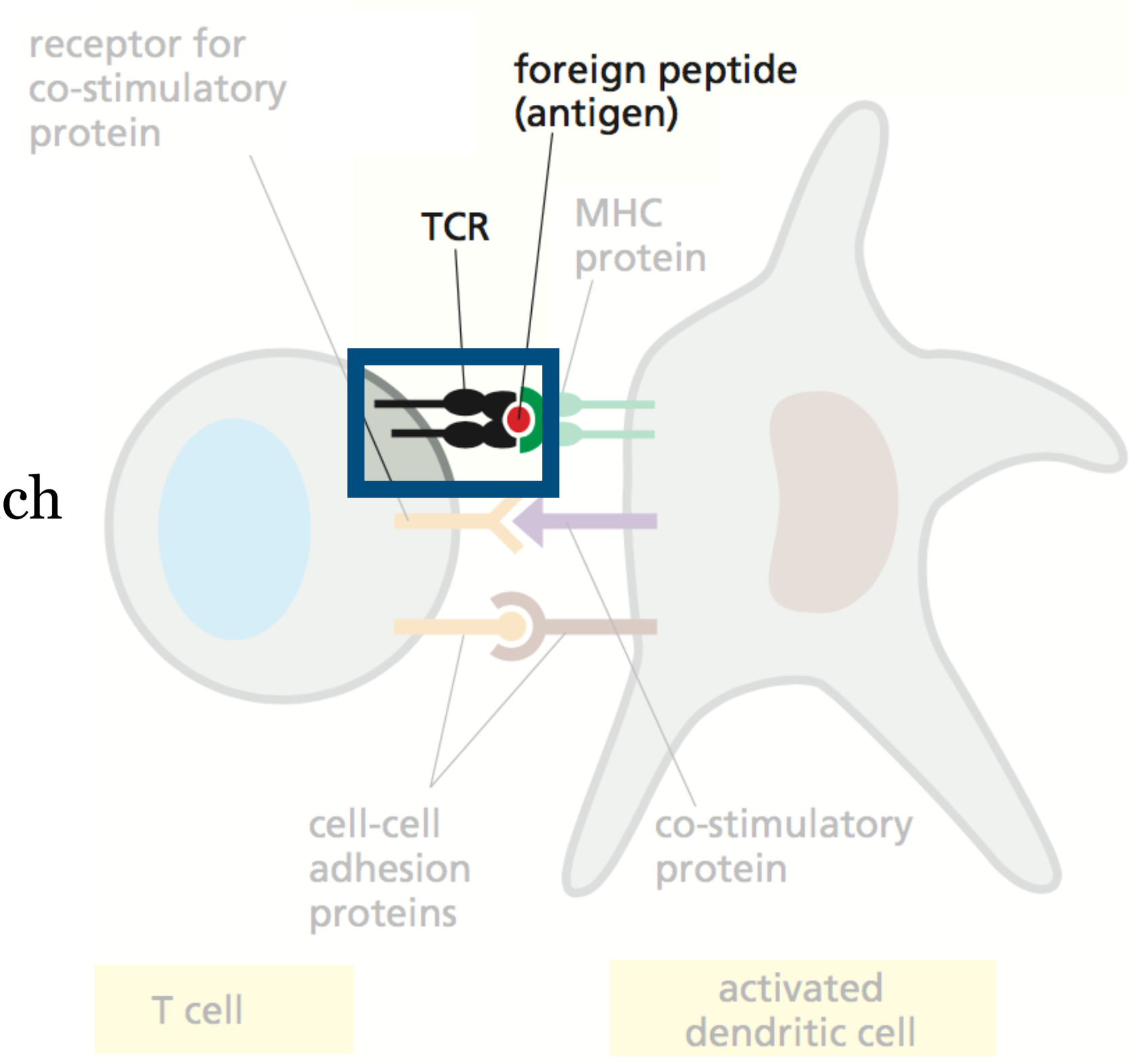
Gaussian processes

Research on TCRs

- Possibilities:
  - Improved diagnostics
    - better understanding of an individual's immune status in different diseases
    - Personalized medicine
      - how a patient would respond to different medications?
- Repertoire level studies (Utilize TCR repertoires of different subjects):
  - e.g. find TCRs associated with some condition
- Sequence level studies
  - e.g. determine epitope specificity of individual TCRs

Goal

- Determine which peptides TCRs recognize

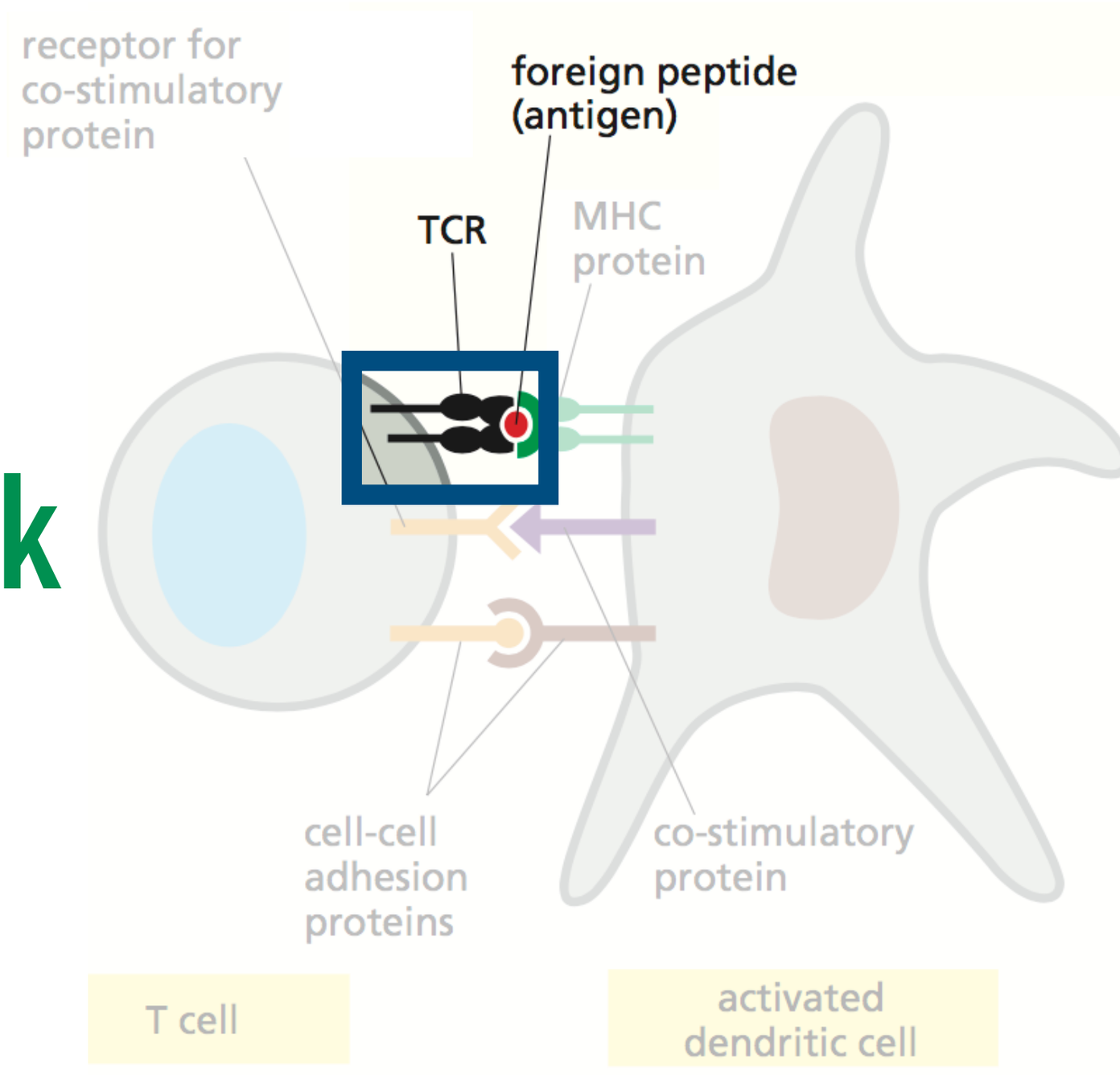


Why machine learning?

- **Perfect solution**:
  - Test experimentally which peptides all possible TCRs ($\sim 10^{18}$) recognize
    - Impossible
- **Machine learning solution**:
  - Assume that similar TCRs behave similarly
  - Based on known TCR-specificities, predict new TCR-specificities (supervised learning)

Supervised learning

- A learning process which looks at annotated data to then automatically annotate similar un-annotated data



Classification task

- Binary classification:
  - Predict wheter a TCR recognizes and binds to a certain peptide or not

Outline

Immunity, T cells and T cell receptors

Motivation and objectives

Sequence data

Kernel methods

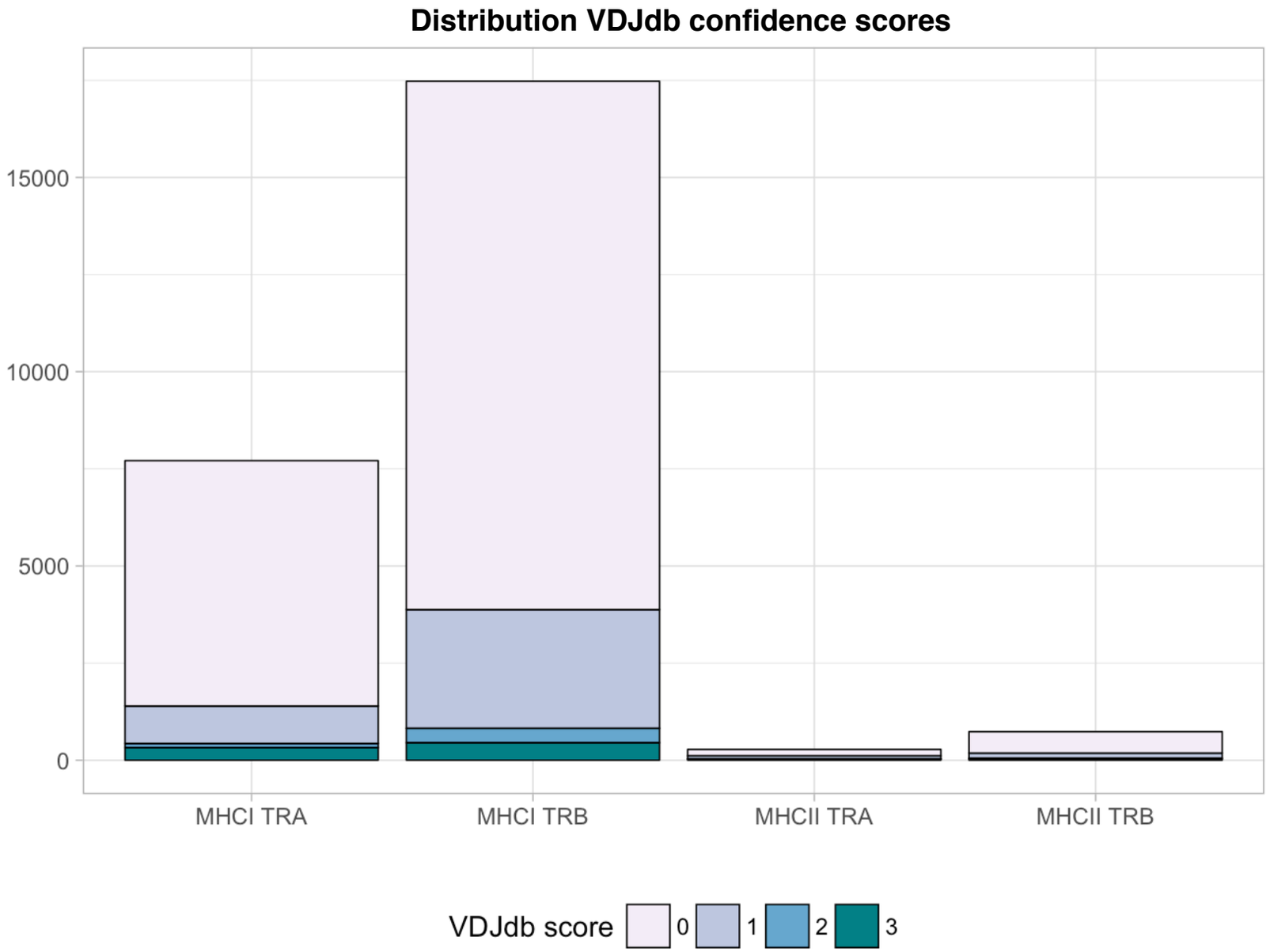
Gaussian processes

Epitope-specific TCRs

- Epitope-specific TCRs can be obtained e.g. from VDJdb

  https://vdjdb.cdr3.net

- TCRs recognizing epitopes from e.g.
  - Infulenza A
  - Cytomegalovirus
  - HIV
  - Epstain Barr Virus

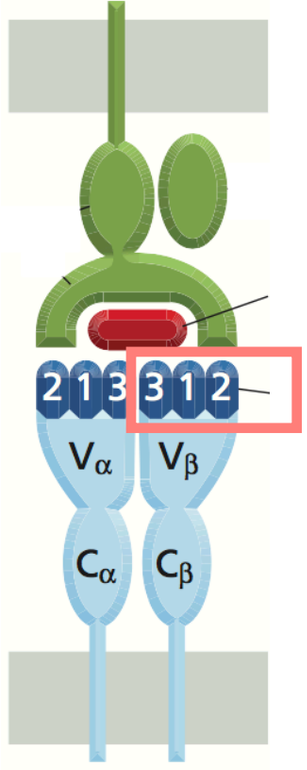


0 - critical information missing, 1 - medium confidence,
2 - high confidence, 3 - very high confidence.

Control sequences

- Negative controls are also needed
- Generally TCRs that recognize an epitope are sequenced, not TCRs that do not recognize that epitope
- We take TCRs that appear only once (singletons) in a subject's TCR reportoire

  ➡ We assume, that these TCRs are unlikely to recognize a certain epitope



TCR amino acid sequences

- Usually a TCR is presented by its CDR3 sequence and V- and J-genes
- CDR1, CDR2 and CDR2.5 are completely determined by V-gene and allele
  - We can construct a table of CDR1, CDR2 and CDR2.5 sequences corresponding to all possible V-genes and alleles
- Examples of TCRβ sequences:

CDR3	CDR1	CDR2	CDR2.5
CASSIQALLTF	SGHDY	FNNNVP	PNASF
CASSVVGGNEQFF	SGDLS	YYNGEE	FPDLH
CASSVAQLAGGTDTQYF	SGDLS	YYNGEE	FPDLH
CSARDPSGLAGGLAETQYF	DFQATT	SNEGSKA	ASLTL

How to utilize sequences?

No alignment
CASSIQALLTF
CASSVVGGNEQFF
CASSVAQLAGGTDTQYF
CSARDPSGLAGGLAETQYF

With alignment
CASSIQ--------ALLTF
CASSVVG------GNEQFF
CASSVAQLA--GGTDTQYF
CSARDPSGLAGGLAETQYF

- Alignment free methods
  - \+ Sequences can have arbitrary lenghts
  - \- Cannot consider position specific information
- Methods that use aligned sequences
  - \+ Can utilize position specific information
  - \+ Can utilize amino acid features (more easily)
  - \- Good alignment can be difficult to get
  - \- New sequences need to be added to the alignment
  - \- New sequences cannot be longer than those in the original alignment

Alignment-free comparisons

- Edit distance: Levenshtein distance
  - Number of changes (insertions, deletions, substitutions) between two sequences:
    - CASSLYF → C**A**AS**S**LY**F** → CAASLYW: distance is 3

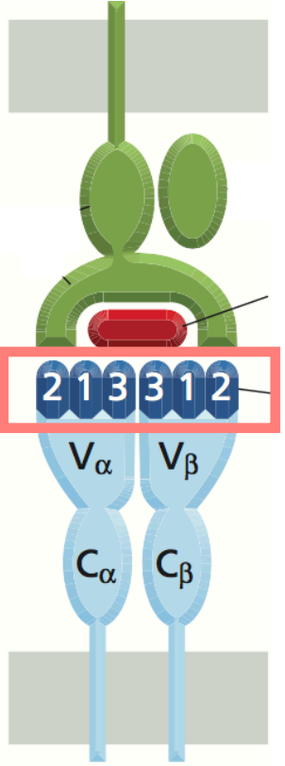
insert delete substitute

- k-mer or motif frequencies
  - Define a set of k-mers,

    all possible or some smaller set

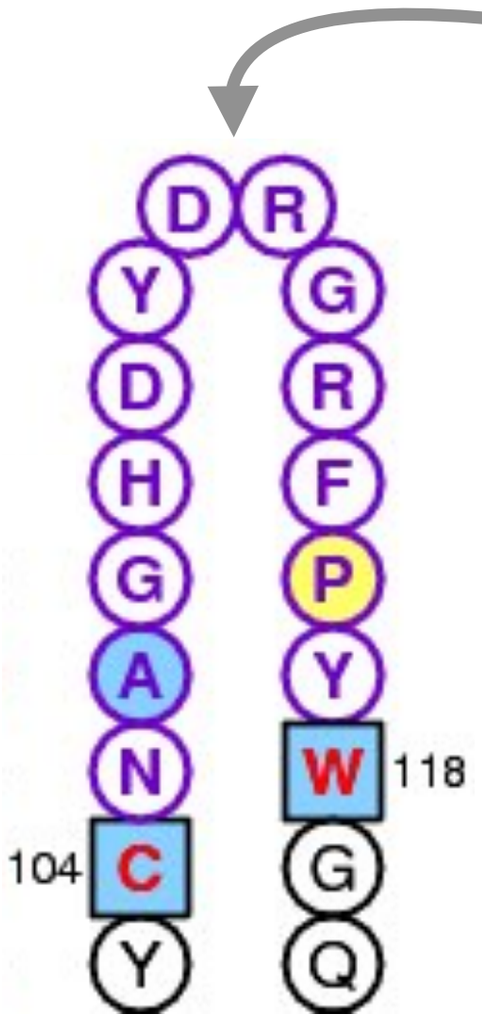
	CAS	ASS	SSL	SLY	⋯
CASSLYFF	1	1	1	1	⋯
CASSIQALLTF	1	1	0	0	⋯
CASSVVGGNEQFF	1	1	0	0	⋯
CAVGDRGYEQYF	0	0	0	0	⋯
⋮	⋮	⋮	⋮	⋮	⋱

- Do not consider similarity between amino acids



Aligning TCR sequences

- There is a limited number of CDR1, CDR2 and CDR2.5 sequences, and we know what they are
  - They can all be aligned according to IMGT definitions
- We assume that CDR3 sequences form simple loops
  - We add gap at the top of the loop for shorter sequences (according to IMGT numbering)
  - Easy to add new sequences to the alignment
- Examples of aligned TCRβ sequences



CDR3	CDR1	CDR2	CDR2.5
CASSIQ--------ALLTF	SGH--------DY	FNN----NVP	P-NASF
CASSVVG------GNEQFF	SGD--------LS	YYN----GEE	F-PDLH
CASSVAQLA--GGTDTQYF	SGD--------LS	YYN----GEE	F-PDLH
CSARDPSGLAGGLAETQYF	DFQ------ATT	SNEG---SKA	A-SLTL

One-hot encoding

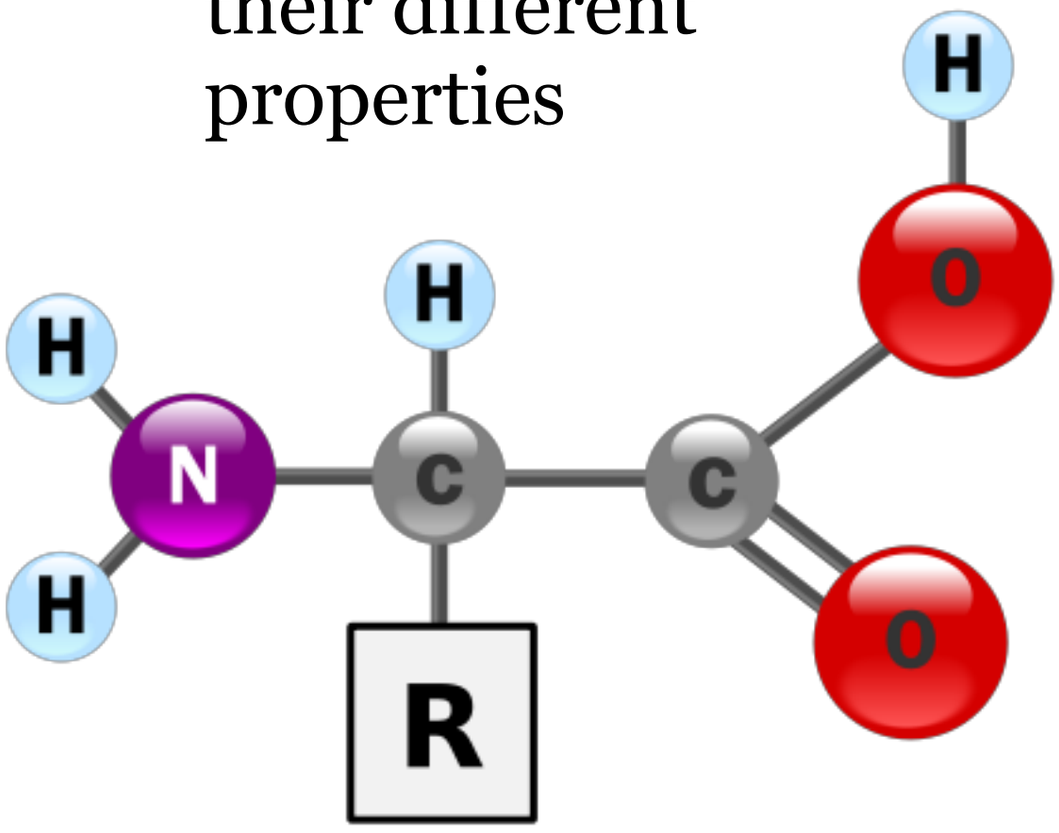
- Most simple numeric presentation
  - Sequences as vectors with constant length
  - Does not consider similarity between amino acids

A	R	N	D	C	E	Q	G	H	I	L	K	M	F	P	S	T	W	Y	V	-
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

C	A	S	S	-	-	-	L	Y	F	F
0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0

Amino acid properties

- There are 20 naturally occurring amino acids
  - R-groups (or side chains) determine their different properties

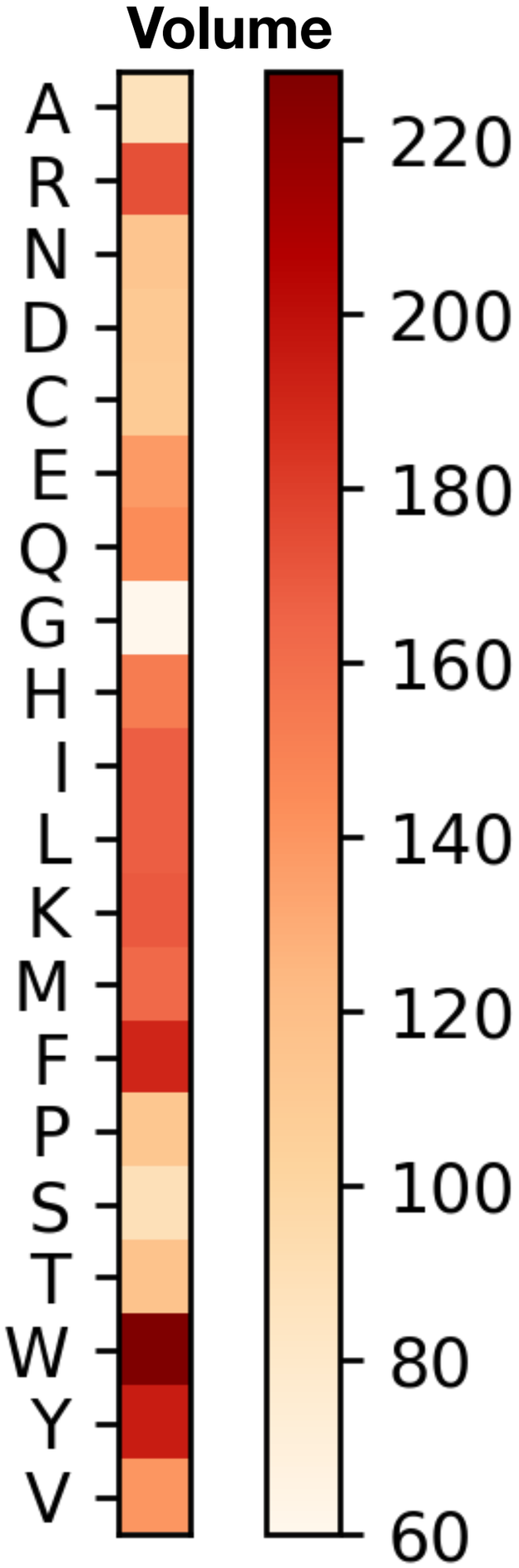


Amino acid	Abbreviation		Chemical	Volume	Hydropathy
Alanine	Ala	A	alipahatic	87	hydrophobic
Arginine	Arg	R	basic	173	hydrophilic
Aspargine	Asn	N	amide	114	hydrophilic
Aspartic acid	Asp	D	acid	111	hydrophilic
Cysteine	Cys	C	sulfur	109	hydrophobic
Glutamic acid	Glu	E	acid	138	hydrophilic
Glutamine	Gln	Q	amide	144	hydrophilic
Glysine	Gly	G	aliphatic	60	neutral
Histidine	His	H	basic	153	neutral
Isoleucine	Ile	I	alipahatic	167	hydrophobic
Leucine	Leu	L	alipahatic	167	hydrophobic
Lycine	Lys	K	basic	169	hydrophilic
Methionine	Met	M	sulfur	163	hydrophobic
Phenyalanine	Phe	F	aromatic	190	hydrophobic
Proline	Pro	P	Cyclic	113	neutral
Serine	Ser	S	hydroxyl	89	neutral
Threonine	Thr	T	hydroxyl	116	neutral
Tryptohophan	Trp	W	aromatic	228	hydrophobic
Tyrosine	Tyr	Y	aromatic	194	neutral
Valine	Val	V	alipahatic	140	hydrophobic

Feature presentation

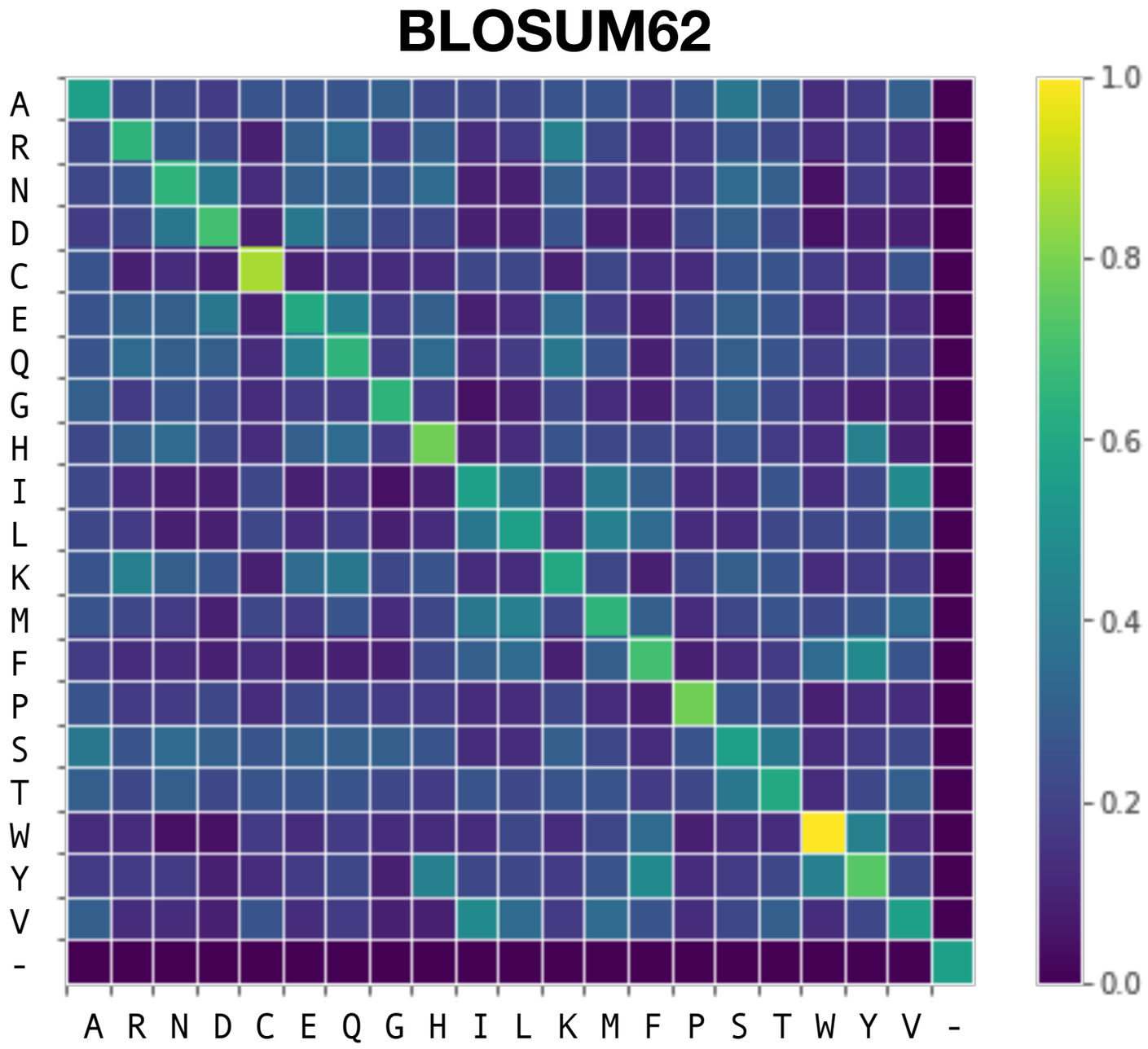
- Use the different amino acid properties as features
  - Concatenate them to make feature vectors for each amino acid, e.g.

$$\begin{bmatrix} \text{volume} \\ \text{charge} \\ \text{hydrophobicity} \\ \text{polarity} \end{bmatrix}$$



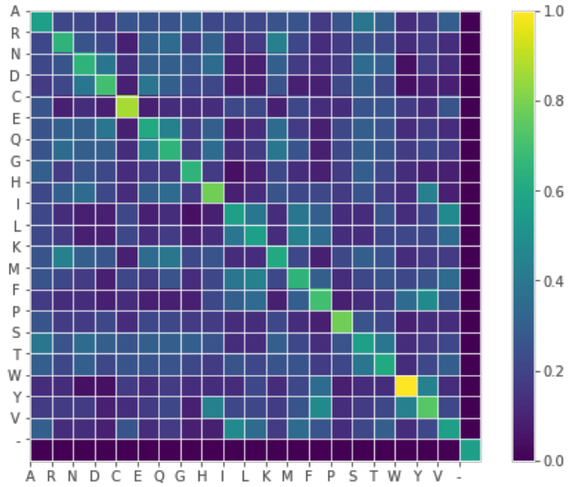
Substitution matrices

- Describe how easily an amino acid can be substitued with another
- Can be based e.g. on:
  - Sequence comparison
  - Sequence comparison by protein blocks
  - Chemical similarity
  - Structural or physical similarity



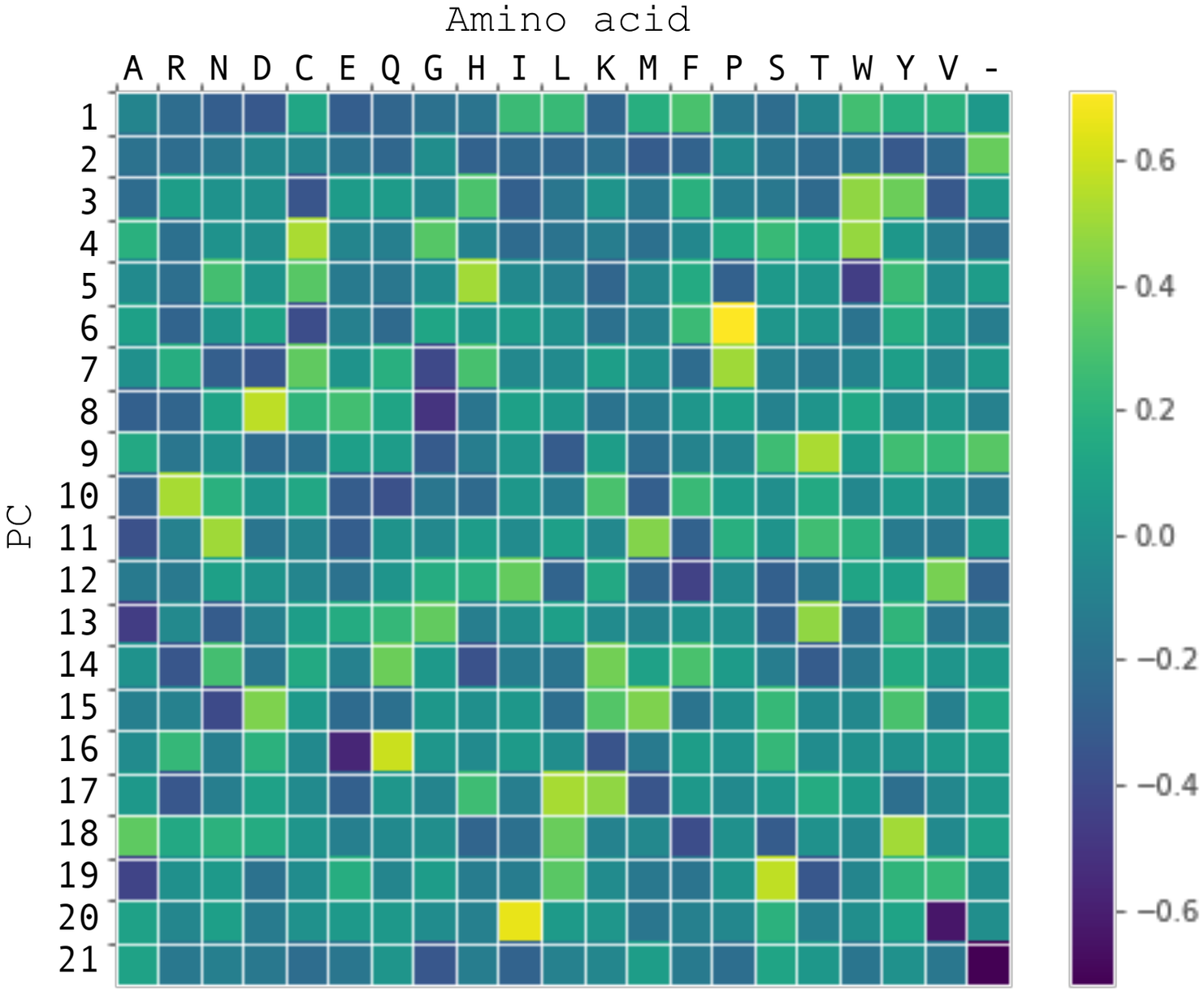
With added gap (-) and scaled into range [0,1]

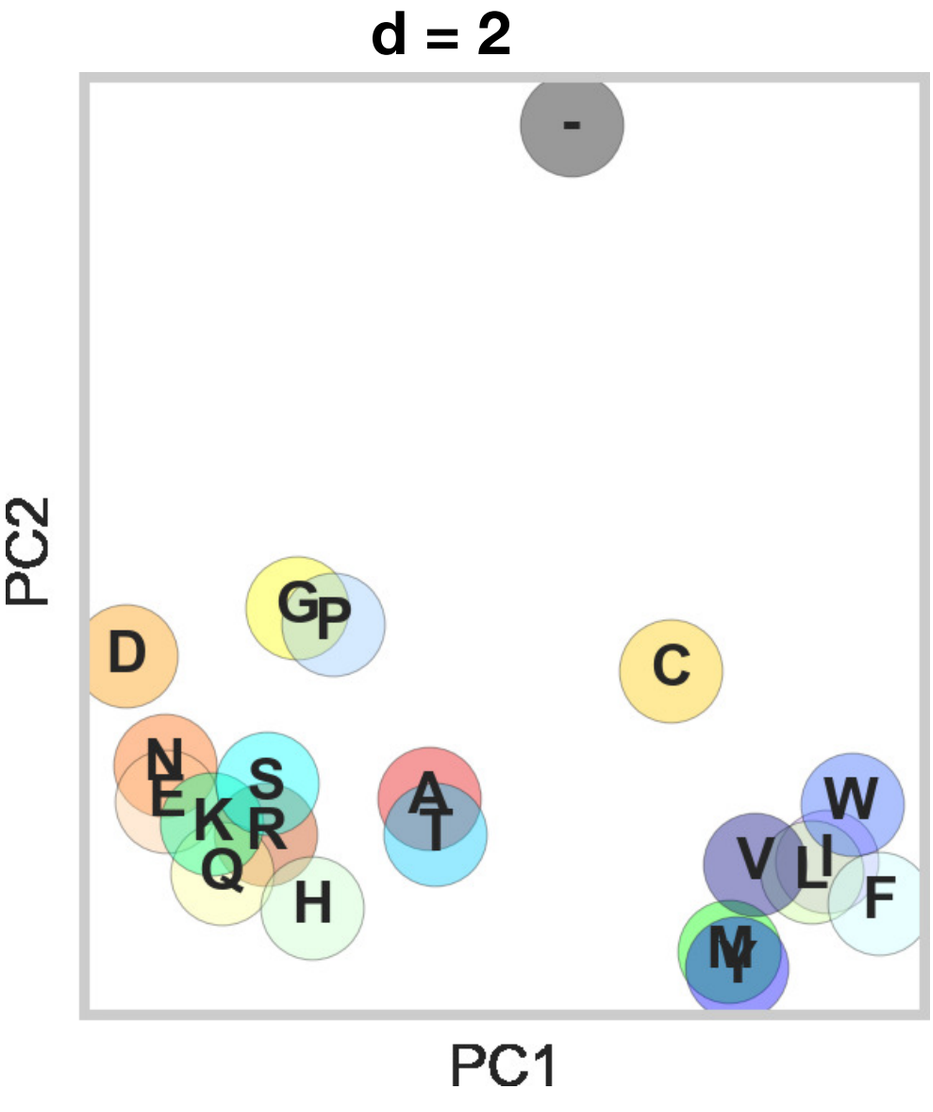
Amino acid features with BLOSUM62

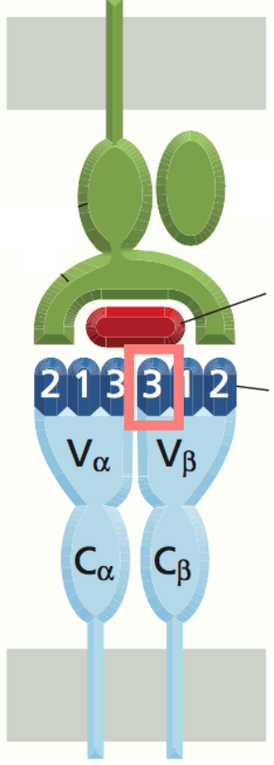


PCA of BLOSUM62

→ feature vectors (size: *d*×1*)* for each amino acid

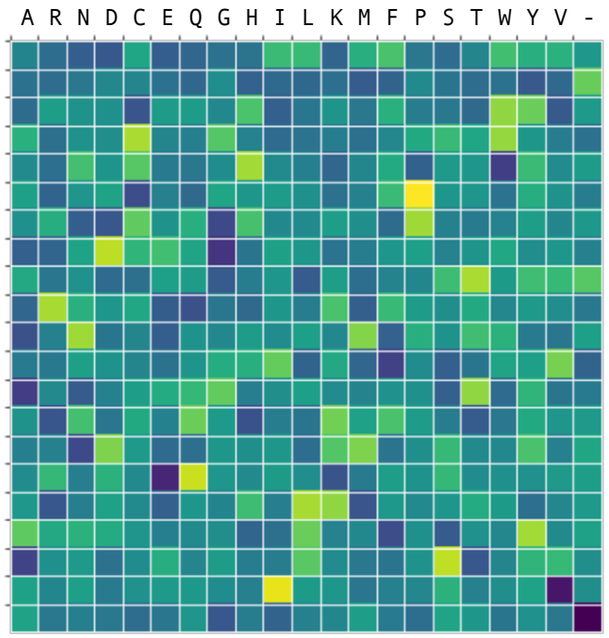




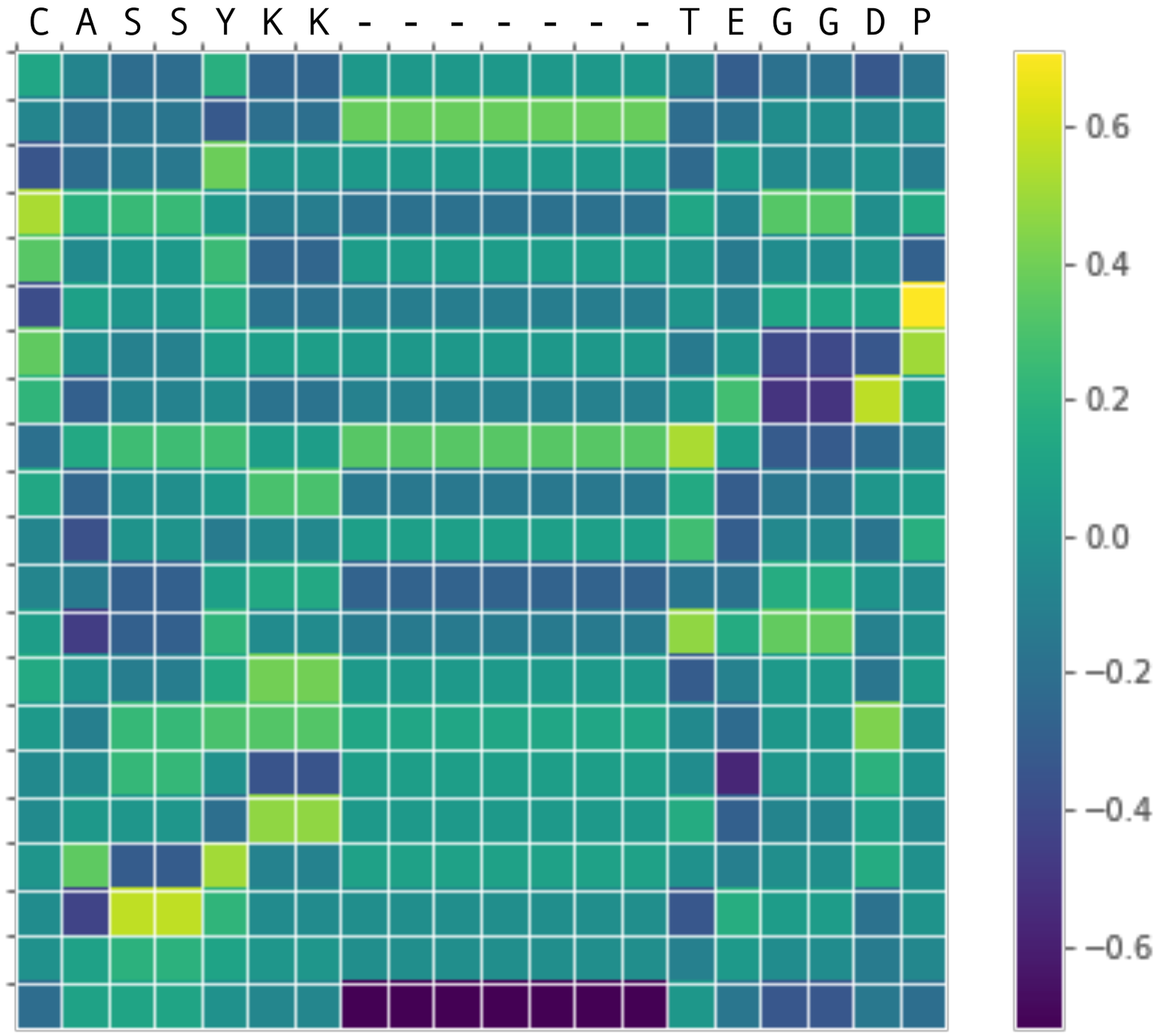


CDR3 presentation with BLOSUM62

PCA of BLOSUM62
→ feature vectors (size: *d*×1)
for each amino acid



Sequence presentation (size: *l*×*d* or (*l*·*d*)×1)



Outline

Immunity, T cells and T cell receptors

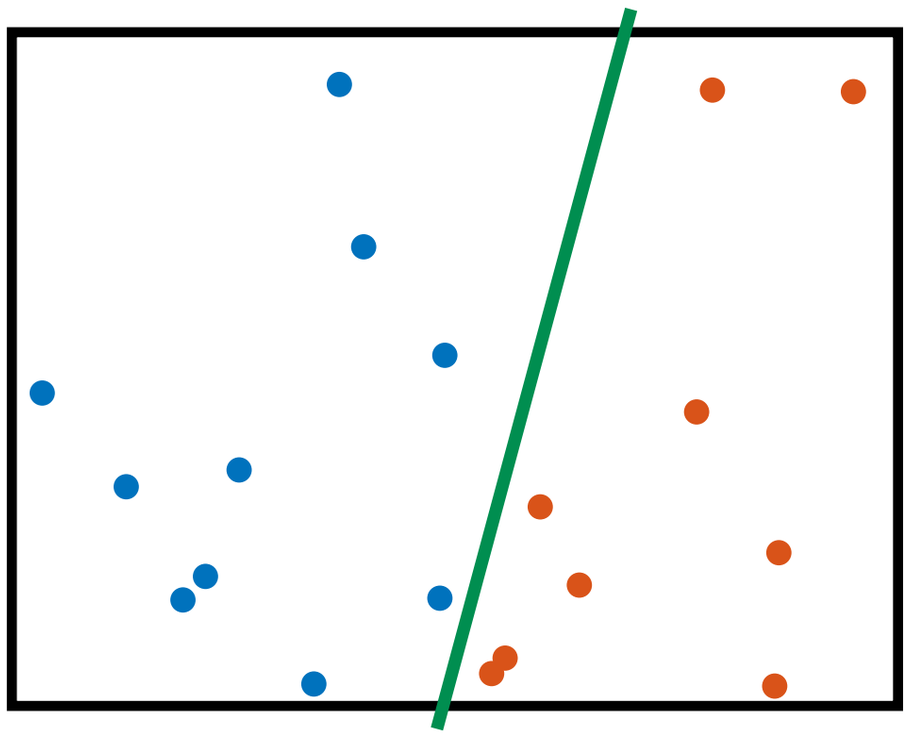
Motivation and objectives

Sequence data

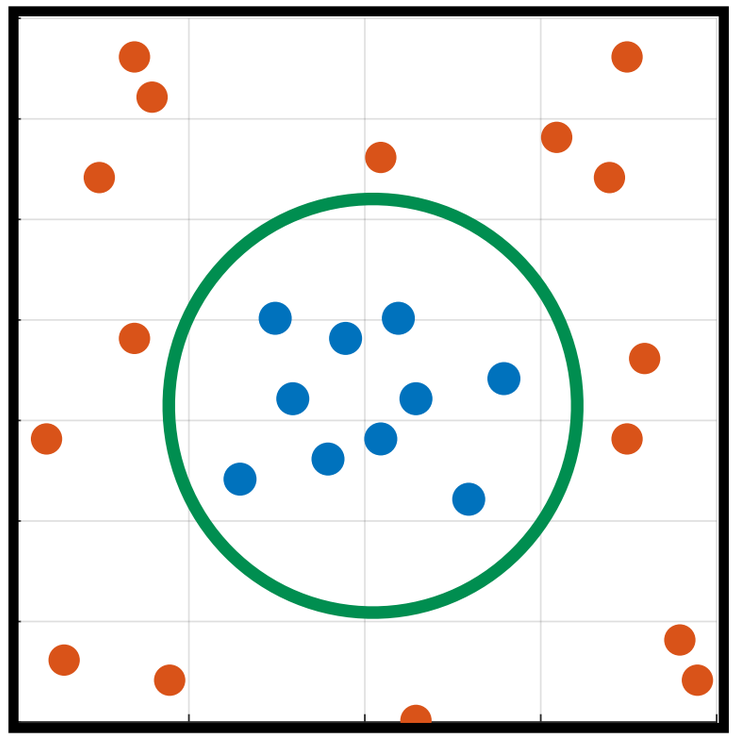
Kernel methods

Gaussian processes

Classification



- Linear classification
  - Fairly simple

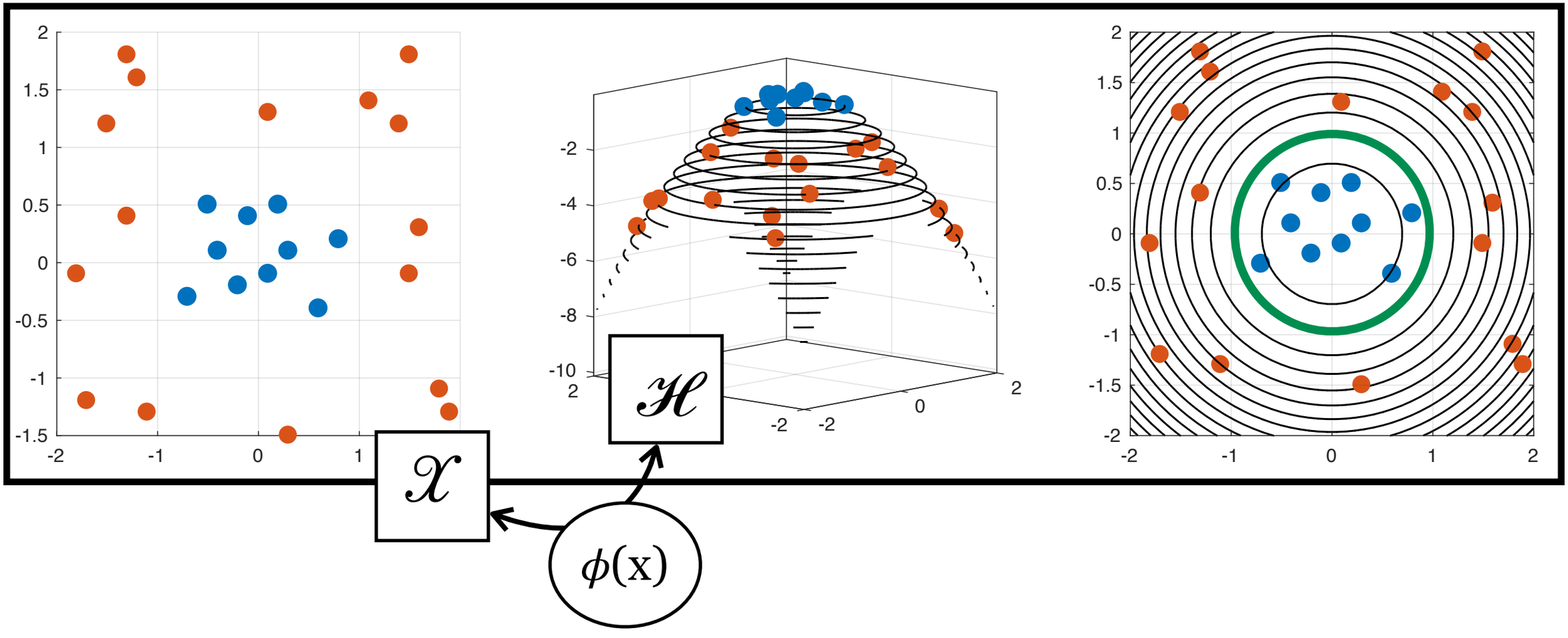


- Non-linear classification
  - More difficult

➡ We utilize kernels

Kernels (1/3)

- Kernel functions allow us to encode the similarity of TCRs
- Kernels can map data $x \in \mathcal{X}$ to a higher dimensional space $\mathcal{H}$, where it is linearly separable



Kernels (2/3)

- Definition:

For a non-empty set $\mathcal{X}$, a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there exists a Hilbert space $\mathcal{H}$ and a function $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}, k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$

- A commonly used kernel is Gaussian kernel (or radial basis function (RBF) or squared exponential (SE)):

$$k(\mathbf{x}, \mathbf{x}'|\theta) = \sigma^2 \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')}{2\ell^2}\right),$$

where ℓ is the length-scale parameter, σ^2 is the magnitude parameter and $\theta = (\ell, \sigma^2)$.

Kernels (3/3)

- Examples of kernel functions:

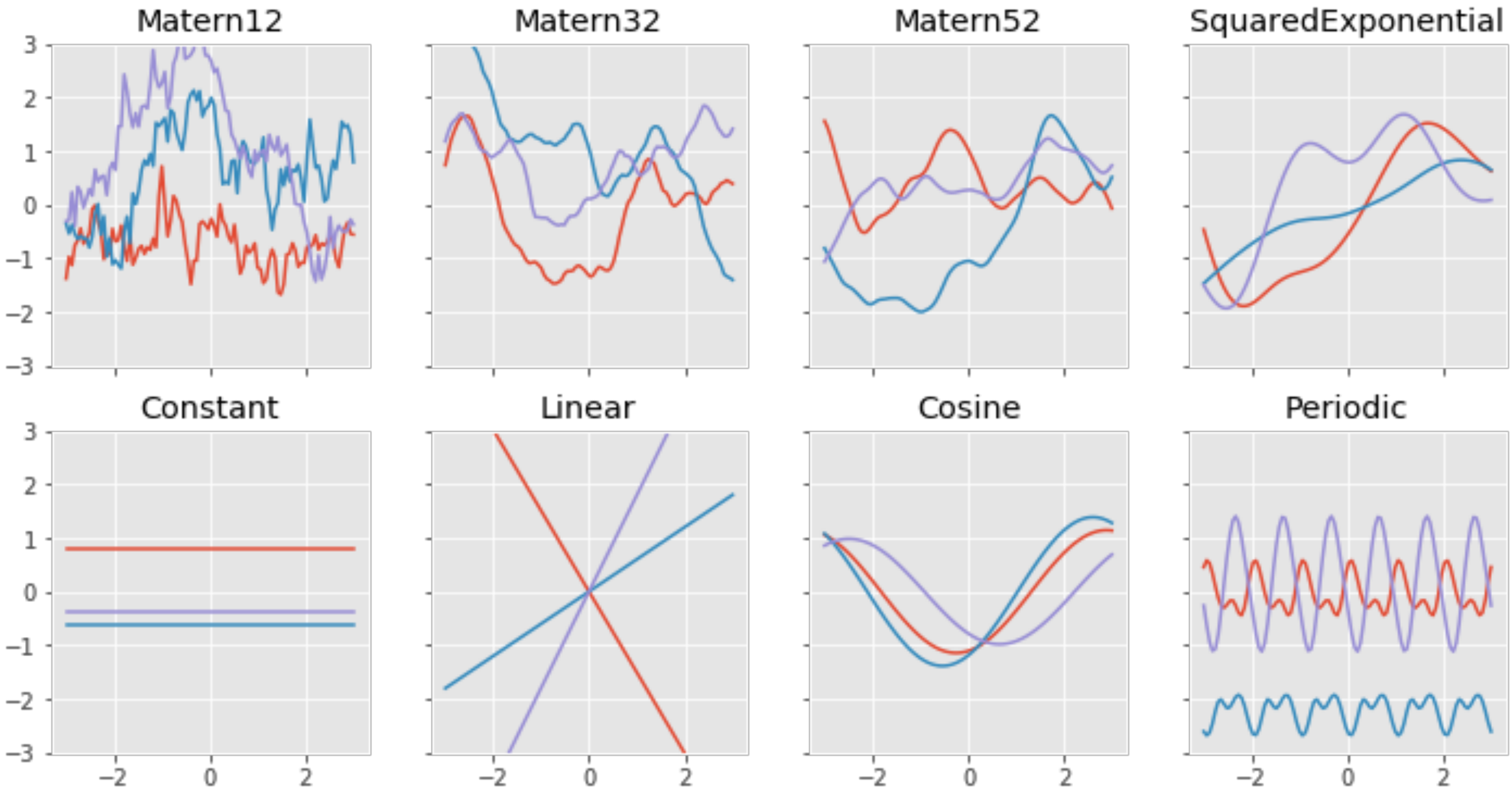


Figure: https://gpflow.readthedocs.io/en/develop/notebooks/kernels.html

Multiple kernel learning (MKL)

- Utilizing multiple kernels allows use to make use of multiple information sources
  - Combine multiple base kernels to obtain one optimized kernel, e.g. as a weight sum of the base kernels
  - In our model, we can utilize the different CDRs from both α and β chains:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\substack{r\in \\ \{1,2,2.5,3\}}} \sum_{\substack{c\in \\ \{\alpha,\beta\}}} w_{cr} k_{cr}(\mathbf{x}, \mathbf{x}'; \theta_{cr})$$

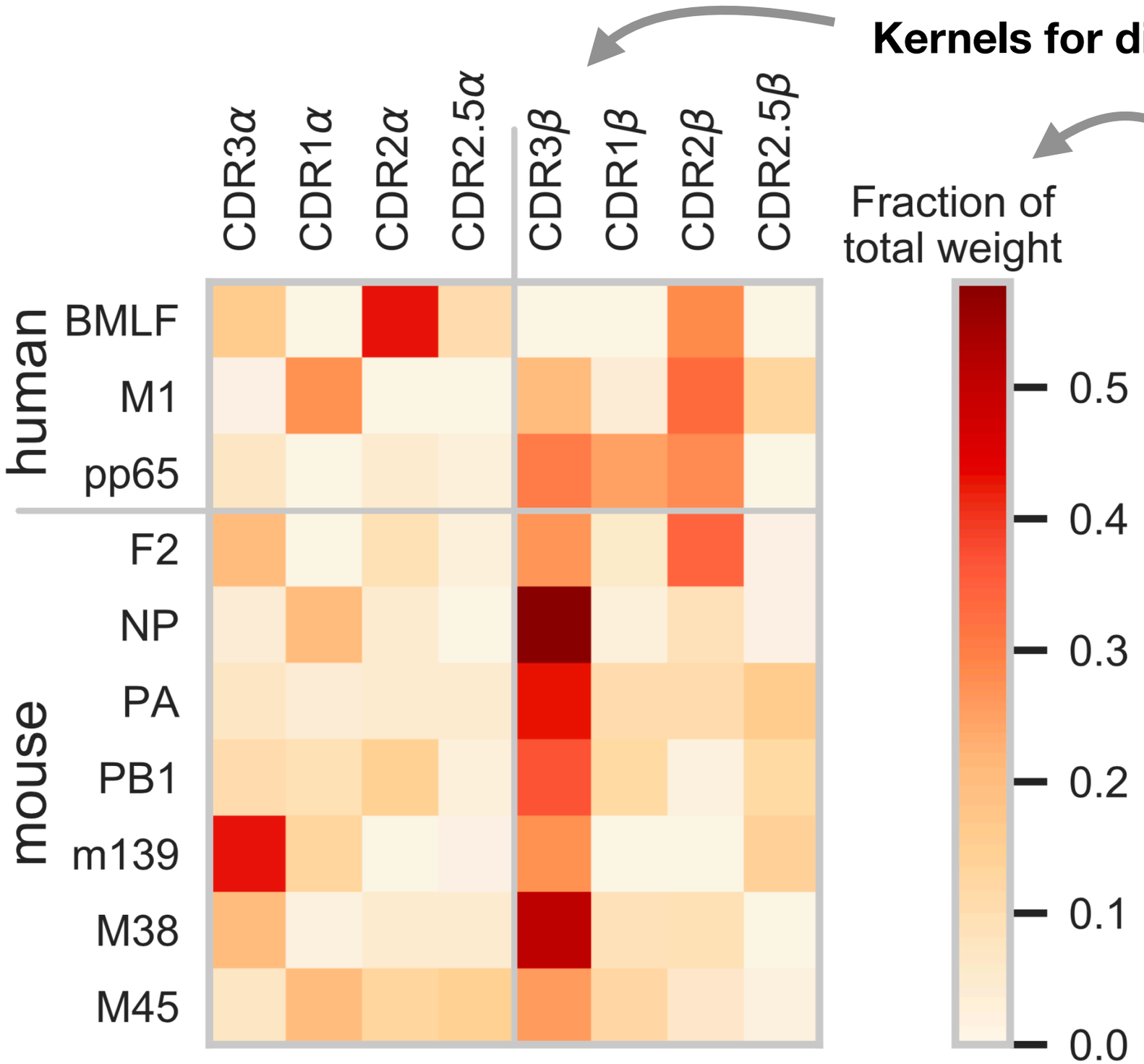
  - We use one-stage-MKL, where we simultaneously optimize the kernel and model

Multiple kernel learning (MKL)

Separate models for different peptides (1 line = 1 model)

Kernels for different CDRs

Optimized weights for the kernels



Outline

Immunity, T cells and T cell receptors

Motivation and objectives

Sequence data

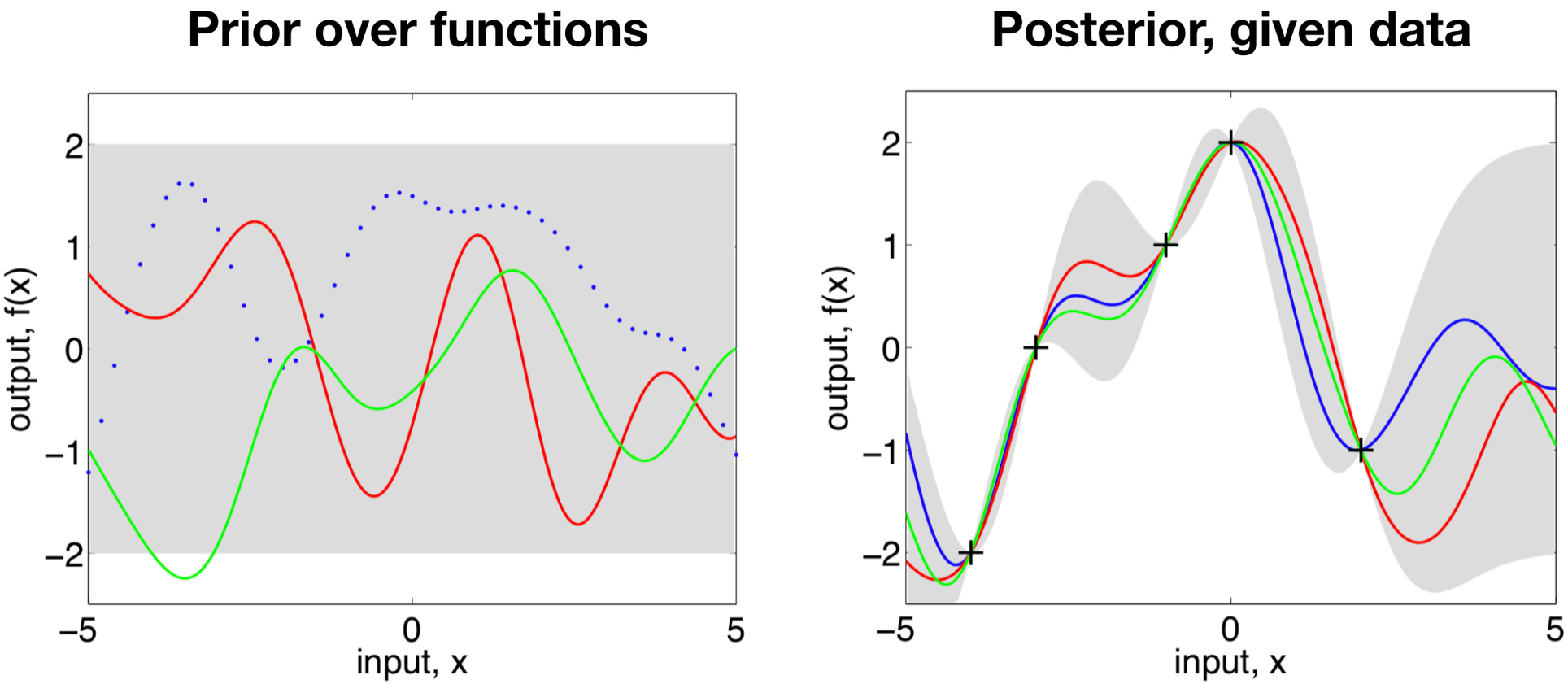
Kernel methods

Gaussian processes

Gaussian processes

- Non-parametric bayesian methods
  - Gives a prior over functions and posterior given data
- Advantages:
  - Able to model complex data
    - without defining specific functions
  - Generally not very prone to overfitting
    - even when there's little data
  - Can also scale to very large data sets
    - When utilizing inducing points
- The standard text book is *Gaussian processes for machine learning*, by Rasmussen & Williams, MIT press, 2006
  - Freely available at http://gaussianprocess.org

GP regression example (1/2)

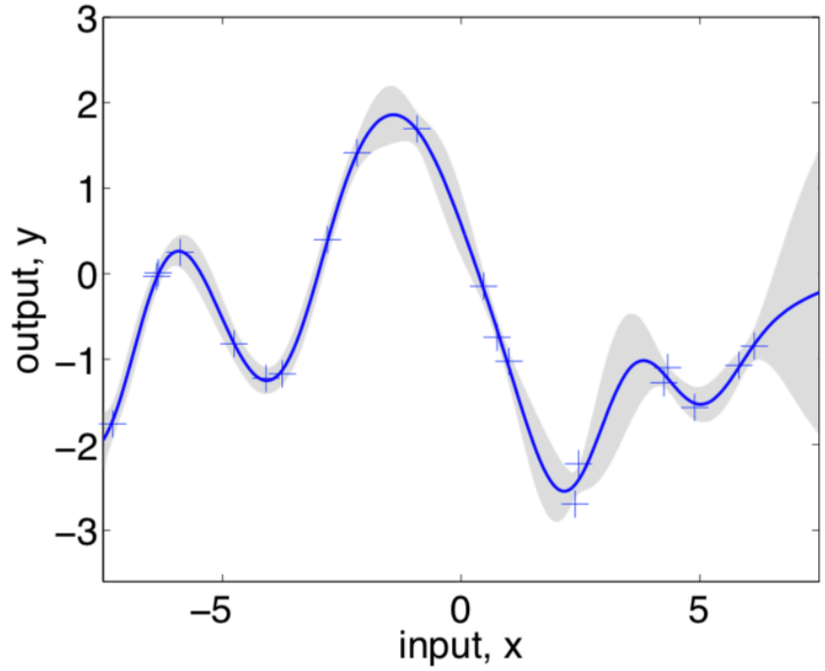


(a), prior (b), posterior

Figure 2.2: Panel (a) shows three functions drawn at random from a GP prior; the dots indicate values of y actually generated; the two other functions have (less correctly) been drawn as lines by joining a large number of evaluated points. Panel (b) shows three random functions drawn from the posterior, i.e. the prior conditioned on the five noise free observations indicated. In both plots the shaded area represents the pointwise mean plus and minus two times the standard deviation for each input value (corresponding to the 95% confidence region), for the prior and posterior respectively.

Figure: [4]

GP regression example 2/2)



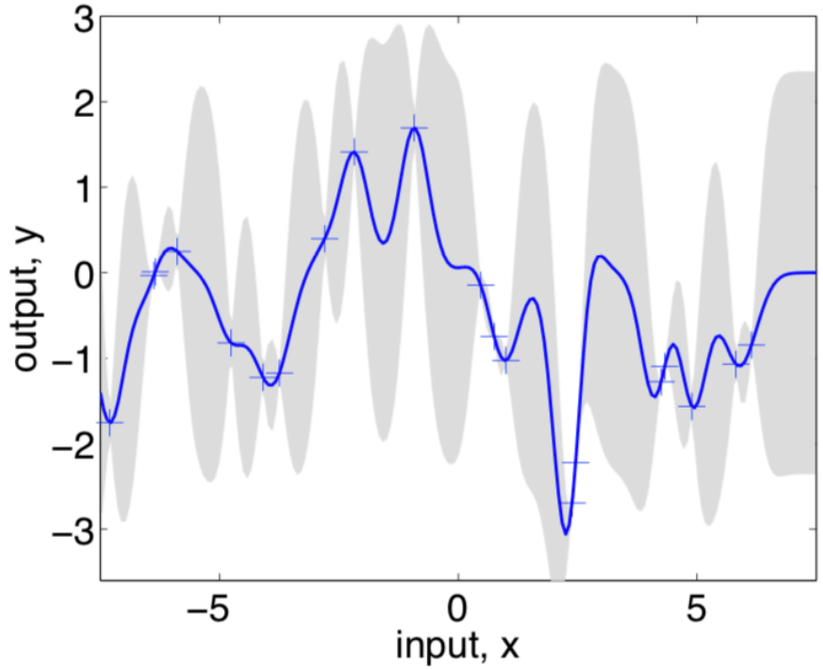
(a), $\ell = 1$

- GPs are very flexible, parameter optimization is important

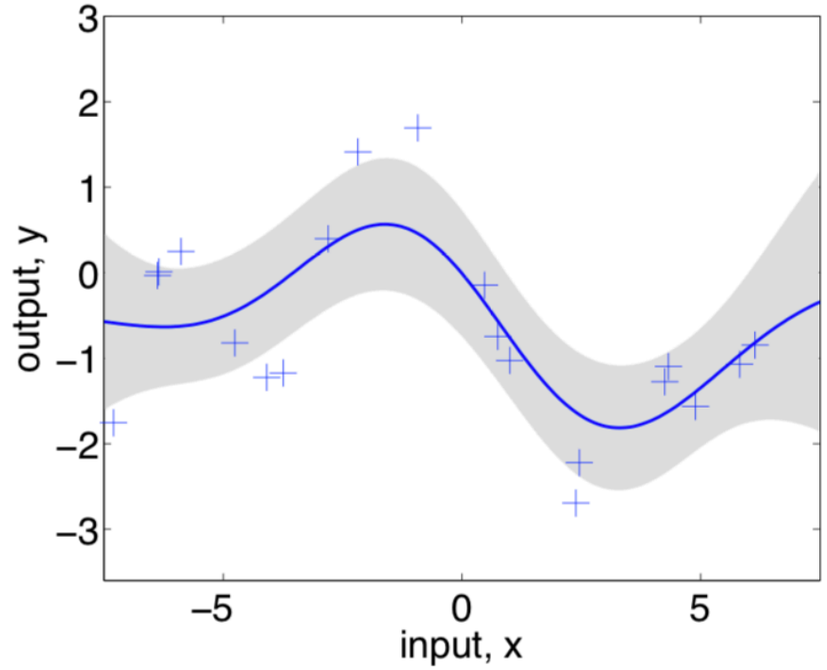
(a) Optimal parameters

(b) Too small lengthscale

(c) too large lengthscale



(b), $\ell = 0.3$



(c), $\ell = 3$

Figure: [4]

Figure 2.5: (a) Data is generated from a GP with hyperparameters $(\ell, \sigma_f, \sigma_n)$ = $(1, 1, 0.1)$, as shown by the + symbols. Using Gaussian process prediction with thes hyperparameters we obtain a 95% confidence region for the underlying function (shown in grey). Panels (b) and (c) again show the 95% confidence region, but th time for hyperparameter values $(0.3, 1.08, 0.00005)$ and $(3.0, 1.16, 0.89)$ respectively.

Defining GPs (1/2)

A stochastic process $f(x) : x \in \mathcal{X}$ is a GP if for any index set $(x_1, \ldots, x_N)$, $f(x_1), \ldots, f(x_N)$ follows N-variate normal $\mathcal{N}(\mathbf{m}, K)$, where $\mathbf{m} \in \mathbb{R}^N$ and $K \in \mathbb{R}^{N \times N}$ is positive semidefinite

- Two values $f(x)$ and $f(x')$ have normal covariance $K(x, x')$
  - Form of $f(\cdot)$ explicitly unknown, we only assume that values covary
- A GP is a distribution $p\left(f(\mathbf{x})\right)$ over function values $f(\mathbf{x}) \in \mathbb{R}^N$, where $\mathbf{x} \in \mathbb{R}^N$

Defining GPs (2/2)

- Let $f(x)$ be modelled as a process $f : \mathbb{R}^1 \to \mathbb{R}^1$
- $f(x)$ is a Gaussian process over x, we denote

$$f(x) \sim \mathcal{GP}\left(m(x), K(x, x')\right),$$

- for any pair $x, x' \in \mathbb{R}^1$ the output covariance equals input covariance,

$$\mathbf{cov}[f(x), f(x')] = K(x, x')$$

- for any subset $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_N) \end{bmatrix} \right)$$

Optimizing GPs

- Model parameters can be optimised by maximising marginal likelihood
- Marginal likelihood for Gaussian likelihood:

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) &= \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\boldsymbol{\theta}) \mathrm{d}\mathbf{f} \\ &= \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}) \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \mathrm{d}\mathbf{f} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}) \end{aligned}$$

- Marginal log-likelihood:

$$\begin{aligned} \log p(\mathbf{y}|\boldsymbol{\theta}) &= \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}) \\ &= \log(2\pi)^{-\frac{N}{2}} |\sigma^2\mathbf{I} + \mathbf{K}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}^T(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{y}\right) \\ &= \underbrace{\frac{N}{2}\log(2\pi)}_{\text{Constant}} - \underbrace{\frac{1}{2}\log|\sigma^2\mathbf{I} + \mathbf{K}|}_{\text{Complexity penalty}} - \underbrace{\frac{1}{2}\mathbf{y}^T(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{y}}_{\text{Datafit}} \end{aligned}$$

GP classification

- Can learn non-linear decision boundaries
  - Learns suitable complexity of the boundary from data
- Models the confindence of the predictions

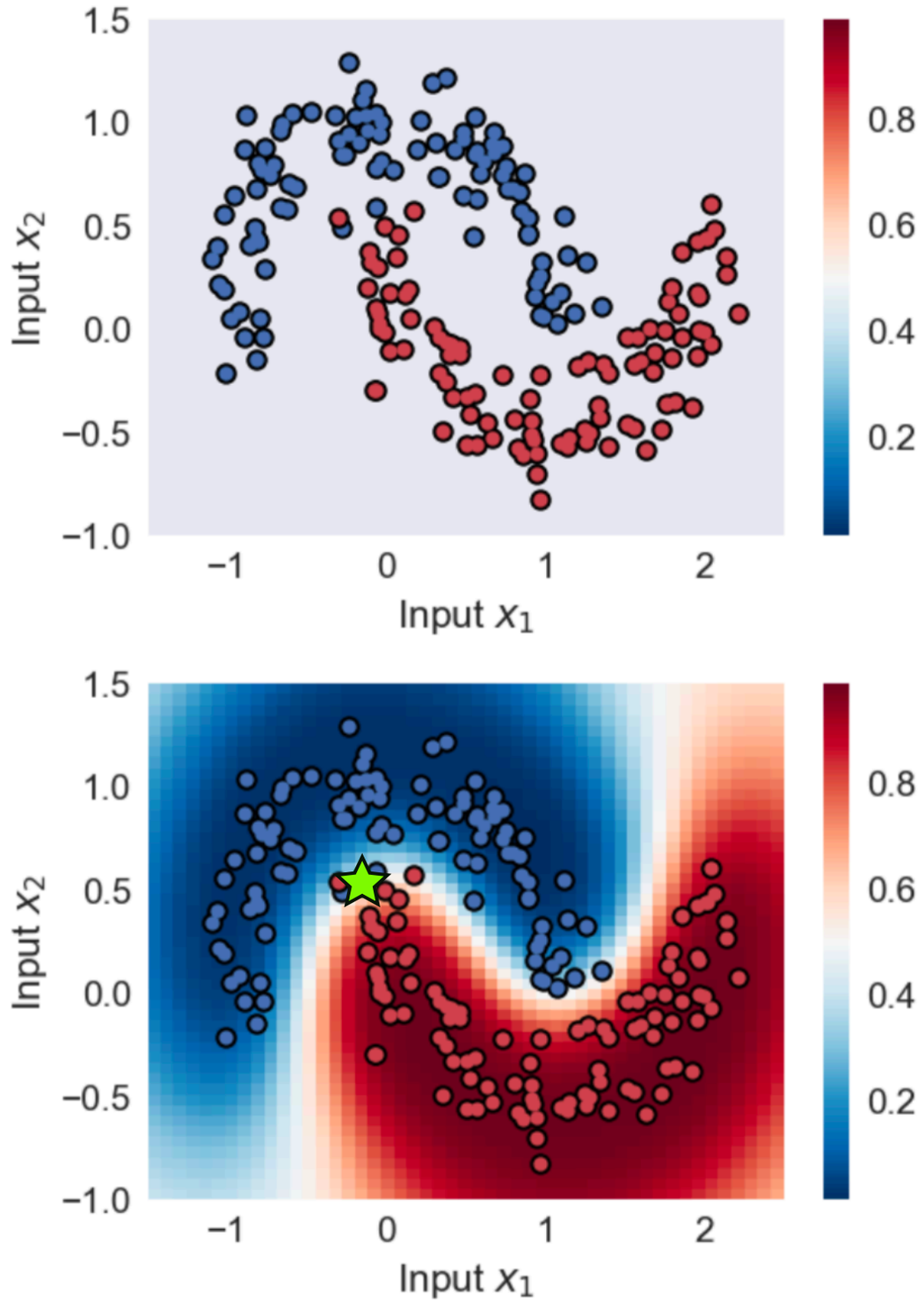


Figure: Michael Riis Andersen, Special course on Gaussian processes, Session 4, 2018

GP classification

- In binary classification, predictions should lie in the interval [0,1]
  - The prior function f is "squashed" to restrict the output to [0,1], e.g. with logistic function:

$$\boldsymbol{\phi}(\mathbf{f}) = (1 + \exp(-\mathbf{f}))^{-1}$$

- Due to the squashing, the computations are not analytically tractable, and some approximations are needed

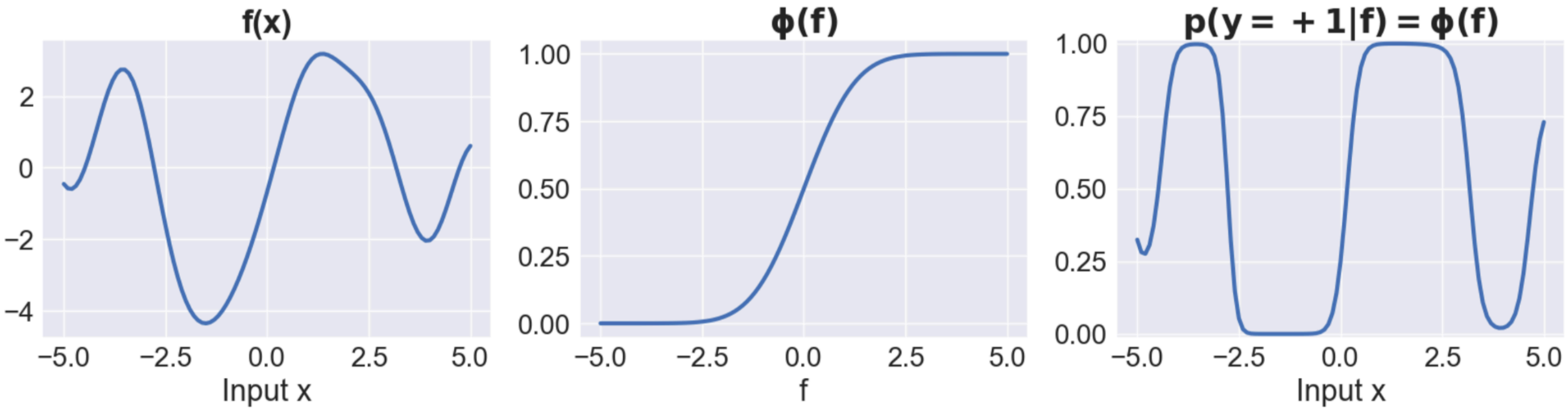


Figure: Michael Riis Andersen, Special course on Gaussian processes, Session 4, 2018

Scalable Gaussian process classification (1/2)

- Computational complexity of GPs scales with $\mathcal{O}(N^3)$, where N is the number of data points
- Instead of conditioning the posterior on all N training points
- Utilize M inducing points

➡ Computation complexity $\mathcal{O}(NM^2)$

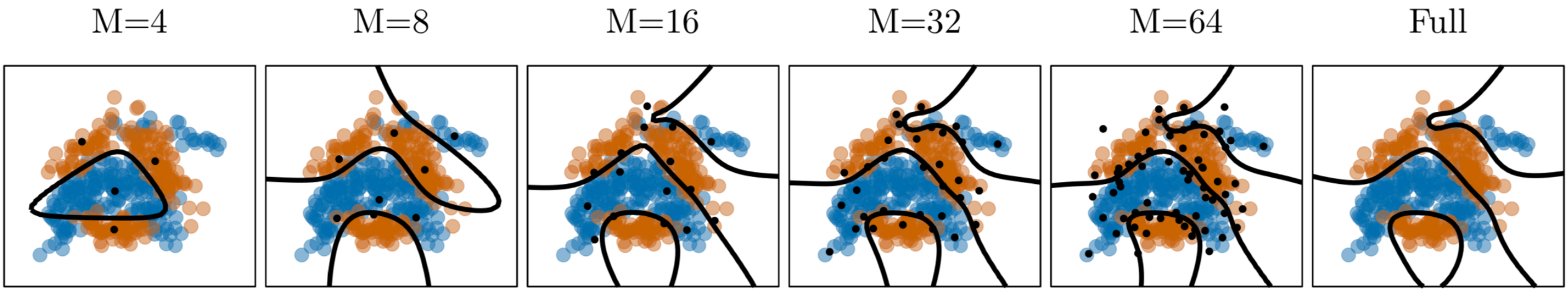


Figure: Scalable variational Gaussian process classification. Hensman, J., et al., 2015.

Scalable Gaussian process classification (2/2)

- We introduce M inducing *landmark* pseudo-sequences $\mathbf{z}_j \in \mathcal{X}$ with associated (label) function values $u_j = f(\mathbf{z}_j) \in \mathbb{R}$.
  - $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)^T$
  - $\mathbf{u} = (u_1, \ldots, u_M)^T$.
- By conditioning the GP with these values we obtain the augmented Gaussian process joint model

$$
\begin{aligned}
p(\mathbf{y}, \mathbf{f}, \mathbf{u}) &= p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \\
p(\mathbf{f}|\mathbf{u}) &= \mathcal{N}(\mathbf{f}|\mathbf{A}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{A}\mathbf{K_{ZZ}}\mathbf{A}^T) \\
p(\mathbf{u}) &= \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}}) \\
\mathbf{A} &= \mathbf{K_{XZ}}\mathbf{K_{ZZ}}^{-1},
\end{aligned}
$$

where $\mathbf{K_{XX}} \in \mathbb{R}^{N \times N}$ is the kernel between observed sequences, $\mathbf{K_{XZ}}$ is between observed and induced sequences and $\mathbf{K_{ZZ}}$ is between induced sequences. The matrix $\mathbf{A}$ projects the M inducing points to the full observation space of N sequences.

Our paper is available at bioRxiv

TCRGP: Determining epitope specificity of T cell receptors

Emmi Jokinen [1,*], Markus Heinonen [1,2], Jani Huuhtanen [3,4], Satu Mustjoki [3,4] and Harri Lähdesmäki [1,*]

[1]Department of Computer Science, Aalto University, Espoo, 02150, Finland. [2]Helsinki Institute for Information Technology, Espoo, 02150, Finland. [3]Translational Immunology Research program and Department of Clinical Chemistry and Hematology, University of Helsinki, Helsinki, 00014, Finland. [4]Hematology Research Unit Helsinki, Helsinki University Hospital Comprehensive Cancer Center, Helsinki, 00014, Finland.

[*] To whom correspondence should be addressed.

Abstract

T cell receptors (TCRs) can recognize various pathogens and consequently start immune responses. TCRs can be sequenced from individuals and methods that can analyze the specificity of the TCRs can help us better understand the individual's immune status in different diseases. We have developed TCRGP, a novel Gaussian process (GP) method that can predict if TCRs recognize certain epitopes. This method can utilize different CDR sequences from both TCRα and TCRβ chains from single-cell data and learn which CDRs are important in recognizing the different epitopes. We have experimented with one previously presented and one new data set and show that TCRGP outperforms other state-of-the-art methods in predicting the epitope specificity of TCRs on both data sets. The software implementation and data sets are available at `https://github.com/emmijokinen/TCRGP`.

Contact: emmi.jokinen@aalto.fi, harri.lahdesmaki@aalto.fi

Other references

[1] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2007). *Molecular biology of the cell*. Garland Science, 5 edition.

[2] Robins, H. S. *et al*. (2009). Comprehensive assessment of T-cell receptor β-chain diversity in αβ T cells. *Blood*, 114(19), 4099–4107.

[3] Lefranc, M. (1999). The IMGT unique numbering for immunoglobulins, T-cell receptors, and Ig-like domains. *Immunologist*, 7(4), 132–136.

[4] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.