



1. Imports

```
import os

from fuse.data.tokenizers.modular_tokenizer.op import ModularTokenizerOp

from mammal.examples.protein_solubility.task import ProteinSolubilityTask
from mammal.keys import CLS_PRED, SCORES
from mammal.model import Mammal
```

2. Model setup

```
# Load Model
model = Mammal.from_pretrained("ibm/biomed.omics.bl.sm.ma-ted-458m.protein_solubi")
model.eval()
```

```
# Load Tokenizer
tokenizer_op = ModularTokenizerOp.from_pretrained("ibm/biomed.omics.bl.sm.ma-ted-458m.protein_solubi")
```

3. Model inference

```
# convert to MAMMAL style
sample_dict = {"protein_seq": protein_seq}
sample_dict = ProteinSolubilityTask.data_preprocessing(
    sample_dict=sample_dict,
    protein_sequence_key="protein_seq",
    tokenizer_op=tokenizer_op,
    device=model.device,
)
```

```
# running in generate mode
batch_dict = model.generate(
    [sample_dict],
    output_scores=True,
    return_dict_in_generate=True,
    max_new_tokens=5,
)
```

```
# Post-process the model's output
ans = ProteinSolubilityTask.process_model_output(
    tokenizer_op=tokenizer_op,
    decoder_output=batch_dict[CLS_PRED][0],
    decoder_output_scores=batch_dict[SCORES][0],
)
```

```
# Print prediction
print(f"{ans=}")
```